



# VoiceNet: Multilingual On-Device Phoneme-To-Audio Alignment

*Kun Jin, Siva Penke, Srinivasa Algubelli*

Samsung Research America, USA

kun.jin@samsung.com

## Abstract

Phoneme-to-Audio Alignment has many applications and is generally considered as an important task in the lip-sync system where an avatar’s lip shape is synchronized with the corresponding speech signal. In this work, we propose such a novel end-to-end on-device multilingual model, VoiceNet, which learns both phoneme recognition and text-independent forced alignment. VoiceNet supports on-device inference in Real-Time as well as in Non Real-time. Moreover, in the Non-RealTime scenario, we show that the performance can be further enhanced when text is given. Our experiments demonstrate competitive performance of VoiceNet compared with state-of-the-art Phoneme Recognition and Forced Alignment results on LibriSpeech and multilingual dataset. Benchmarked on a set of Galaxy phone devices, VoiceNet achieves the average phoneme inference latency of 6ms on CPU, demonstrating the high computational efficiency. Furthermore, VoiceNet can achieve 2x speedup on GPU and 10x speedup on NPU.

**Index Terms:** multilingual phoneme recognition, forced alignment, on-device inference

## 1. Introduction

Phoneme-to-audio alignment aims to determine the precise timing of phonemes as they are spoken in an audio signal. This task is essential for a wide range of speech and phonetic areas, including speech analysis of phonetic and prosodic features, automatic subtitle generation, and speech synthesis. One significant application of phoneme-to-audio alignment is lip synchronization (lip-sync), which ensures that visual articulatory movements match input speech soundtrack as well as transcript [1], and has been actively researched since the early 1970s. This technology gained increasing importance in virtual reality (VR), augmented reality (AR), and digital avatars, where naturally and precisely aligned speech animation enhance user immersion [2].

Beyond lip-sync, phoneme recognition is also fundamental to Automatic Speech Recognition (ASR). Recent advances in ASR have been driven by Transformer-based models [3, 4] and Convolutional Neural Networks (CNNs) [5, 7], which have outperformed conventional methods, such as RNN-Transducer [8, 9] and Hidden Markov models (HMM) [10]. While these modern architectures achieve state-of-the-art results, their deployment on mobile devices is hindered by high resource demands—they generally require a large number of model parameters, consume significant memory, and incur high inference latency. The challenge intensifies when considering the need for phoneme recognition that integrates seamlessly with forced alignment, a feature critical for lip-sync or enhancing user interaction in real-time applications. Current ASR models optimized

for mobile devices [11, 12] make strides toward efficiency but still lack a robust solution for on-device phoneme recognition with forced alignment capabilities.

To bridge this gap, we propose VoiceNet, a novel end-to-end multilingual convolutional neural network model designed specifically for mobile devices. VoiceNet not only achieves accurate real-time phoneme-to-audio alignment but also incorporates an advanced forced alignment mechanism without the need for additional external models. Composed solely of 1D Convolutions, ReLU, Batch Normalization, Dropout, Residual Connections, and Self-Attention, VoiceNet is highly optimized for both training and inference. This model configuration allows VoiceNet to outperform the current on-device state-of-the-art models by reducing the Phoneme Error Rate (PER) by 3.8% in a multilingual test dataset, while offering much lower inference latency (6ms) and requiring fewer parameters (6.4M).

Our contributions are listed as follows:

- We proposed Voicenet, a real-time on-device end-to-end computationally efficient multilingual phoneme recognition and text-independent forced alignment model.
- Given text, we propose a phoneme-group-to-subword alignment mechanism to further reduce the multilingual PER by 2.3%.
- To our best knowledge, this is the first work to apply on-device text-dependent phoneme-to-audio alignment without extra models, for example, Grapheme-to-Phoneme (G2P) model. Especially in the era of digital assistants enabled by Large Language Model (LLM) and Text-To-Speech (TTS), our work could have high impact in the interactive lip-sync scenario.
- Through experiments, we demonstrate that VoiceNet outperforms on-device Real-Time RealPRNet model [13] by 6.3% on TIMIT dataset and performs even better than Non-RealTime multilingual Conformer-small [3] phoneme recognition model by 3.8% PER reduction on average with fewer parameters and much lower inference latency. Moreover, given text, VoiceNet could achieve competitive performance compared with public phoneme aligners.

## 2. VoiceNet Architecture

VoiceNet is an end-to-end convolutional neural network which is composed of an Audio Encoder (AE) and a Phoneme Model (PM). AE is to extract 39 mel-filterbank features (MFCCs) calculated from 25ms time windows with a 10ms frame overlap, and PM outputs a probability distribution over phonemes per frame.

## 2.1. Audio Encoder

It has been a trend to extract acoustic features with a pretrained large acoustic model, such as HuBERT [14], XLS-R [15] and Wav2Vec 2.0 (W2V2) [4] which are self-supervised learning (SSL) models capturing rich representations from raw audio. Whisper encoder [16] has also recently gained popularity as a base model for acoustic features extraction. However, they are too computational expensive for on-device use case and are prohibitive for real-time on-device applications. Instead of referring to these audio extraction models, we still harness the conventional MFCC features which are simple but effective, and widely used in automatic speech recognition (ASR) models.

AE is designed to replicate the Mel-Filterbank functions using the neural network layers (*e.g.*, 1D Convolution layers and fully connected layers) to extract MFCCs, which is illustrated in Figure 1. The benefits of AE are first to make use of the operators acceleration of deep neural network framework (*e.g.*, TensorFlow), and second to generalize the deployment of VoiceNet without the dependency on native implementation of MFCC algorithm via JNI call on Android. The input to AE is an audio frame, which is raw audio of 25ms time window, and the output is 39 MFCC features.

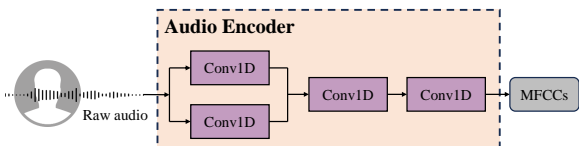


Figure 1: Architecture of Audio Encoder in VoiceNet.

## 2.2. Phoneme Model

Following the recent convolutional ASR models [7, 6], PM employs a block architecture as depicted in Figure 2a. Specifically, Block 1 and Block 3 mirror the architecture illustrated in Figure 2c, including a sequence of following operations: a 1D-Convolution, Batch Norm, ReLU and dropout. Block 2, shown in Figure 2b, consists of two composite blocks, each block is composed of three sub-blocks and each block input is connected directly into the last sub-block via a residual connection. The residual connection is first projected through a  $1 \times 1$  convolution to account for different numbers of input and output channels, then through a batch norm layer. The output of this Batch Norm layer is added to the output of the Squeeze-and-Excitation (SE) layer [5] in the last sub-block. The result of this sum is passed through the activation function and dropout to produce the output of the current composite block. PM has two blocks of Block-2 and an additional self-attention layer to capture long-range dependencies and enhance contextual understanding, as well as two fully connected (FC) layers: one FC layer activation function is softmax function to model a probability distribution over phonemes, the other FC layer activation function is sigmoid function to model the presence of phoneme boundaries. The phonemes are categorized based on the International Phonetic Association (IPA) [17] which is generalized for most of the languages.

Formally, given an audio signal  $X_S \in \mathbb{R}^{1 \times T_{raw}}$  and an annotated phoneme sequence  $Y \in \mathbb{R}^{1 \times N} = \{y_1, y_2, \dots, y_N\}$ , AE is used to encode the audio into representations  $\Phi(X_S)$  (MFCCs) and PM is to learn a neural function  $f$  to predict the phoneme sequence  $\hat{Y} = f(\Phi(X_S)) = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T\}$  where

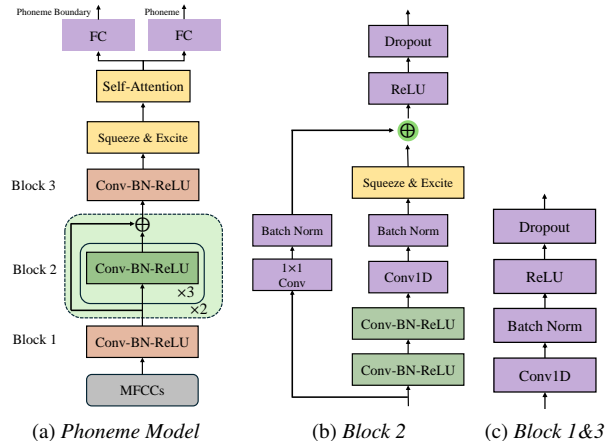


Figure 2: Architecture of Phoneme Model in VoiceNet.

$\hat{y}_i$  is  $i$ -th predicted phoneme,  $N$  is the number of phoneme labels in the text and  $T \geq N$ . The network is trained to minimize a loss function  $\mathcal{L}(Y, \hat{Y})$  defined as follows.

The loss function used in the PM model is composed of three parts: 1) Categorical Cross-Entropy Loss for phoneme recognition  $\mathcal{L}_{CCE}$ ; 2) Binary Cross-Entropy loss  $\mathcal{L}_{BCE}$  for phoneme-boundary classification which encourages the sharp transition between different phonemes and is important for forced alignment task; 3) Poly-1 Loss [18] is used to mitigate class imbalance and is defined as  $\mathcal{L}_{Poly-1} = \mathcal{L}_{CCE} + \epsilon(1 - P_i)$  where  $P_i$  stands for the prediction probability for the target label and  $\epsilon$  is the coefficient which is a hyperparameter. In the phoneme recognition task, given a speech corpus, some phoneme classes are more common than others, leading to class imbalance issue [19]. The Poly-1 Loss is to penalize those uncertain predicted labels with low probability which is usually the case of minor classes, and thus mitigate the imbalance issue. Observing that  $\mathcal{L}_{CCE}$  is part of  $\mathcal{L}_{Poly-1}$ , we remove separate  $\mathcal{L}_{CCE}$  to avoid extra hyperparameter. By integrating the above components, we formulate the final loss function as:

$$\mathcal{L} = \mathcal{L}_{Poly-1} + \lambda \mathcal{L}_{BCE} \quad (1)$$

In contrast with automatic speech recognition (ASR) models [7, 6] which are trained with CTC (Connectionist Temporal Classification) loss that includes a blank character, we do not include CTC loss in our loss function because the blank symbol conceals the beginning and end of each phoneme, resulting in inaccurate boundary detection, which is also shown in the experiments.

## 3. Phoneme Recognition Accuracy Improvement with Text

When audio corresponded text is available, a Grapheme-to-Phoneme (G2P) model is usually used to convert words into phonemes which are then aligned with predicted phonemes based on Dynamic Time Warping (DTW) to correct false phoneme prediction. However, from one hand, relatively accurate multilingual G2P model [20] suffers from large model size while light G2P model [21] is limited to high error rate; from the other hand, extra multilingual G2P model needs to be deployed on the mobile device. Moreover, the pronunciation and text may not correspond exactly since it is common for non-native speakers to have false pronunciation for words while language model

based decoder in Automatic Speech Recognition (ASR) usually corrects the words based on the context.

To address these challenges, we extract consistent letter-phoneme pairs extracted from a pronunciation lexicon based on phoneme-based sub-word modeling [22]. Figure 3 illustrates an alignment for the word “SPEECH”. The alignment is represented as a set  $\{(0, 0), (1, 1), (2, 2), (3, 2), (4, 3), (5, 3)\}$  of which each element is a pair of (letter index, phoneme index). In practice, We could have one-to-one (“cat”), many-to-one (“ch”) and even many-to-many alignments. The mechanism [22] is composed of three steps: 1) Using an aligner, `fast_align`, to generate letter-phoneme alignment from a pronunciation dictionary; 2) Extract consistent subword-phoneme pairs from alignment; 3) Refine subword-phoneme pairs based on statistics of them over letter sequences. To simplify the mechanism and generalize to unseen words, the method only generates a list of sub-words and a set of pairs (sub-word,  $\{ph_0, ph_1, \dots\}$ ) with weights. We further split any word with those sub-words.

0	1	2	3	4	5
<b>S</b>	<b>P</b>	<b>E</b>	<b>E</b>	<b>C</b>	<b>H</b>
			/		/
s	p	i		tʃ	
0	1	2	3		

Figure 3: Alignment for the Word “SPEECH”.

Given the transcript of audio and predicted phoneme sequence from VoiceNet model, the goal is to improve phoneme classification accuracy through the alignment between text and predicted phonemes. Formally, given annotated phoneme sequence  $Y = \{y_1, y_2, \dots, y_N\}$  where  $y_i$  is  $i$ -th phoneme derived from text and predicted phoneme sequence  $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T\}$ , where  $\hat{y}_j$  is the  $j$ -th frame predicted phoneme. We seek to align elements in  $\hat{Y}$  with those in  $Y$ . Instead of using G2P model to convert text into phoneme sequence  $Y$ , we construct a phoneme group sequence  $\tilde{Y} = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N\}$  where  $\tilde{y}_j = \{ph_0^j, ph_1^j, \dots, ph_m^j\}$ . For example, sub-word *ch* is corresponding to  $\{ʃ, tʃ, k\}$ . For the sub-word corresponding to combination of phonemes, we add those phonemes separately in the sub-word corresponding phoneme group. From the above mechanism, we first split text into sub-word sequence based on the generated sub-word list, each of which corresponds to a phoneme group  $\tilde{y}_j$ .

DTW is then applied based on PM output probability matrix  $P \in \mathbb{R}^{M \times K}$  to correct the phoneme prediction, where  $M$  is the number of predicted phonemes and  $K$  is the number of distinct phoneme labels.  $P_{i,j}$  denotes the likelihood of the  $i$ -th frame being predicted as label  $PH_j$ . In our case, the predicted phonemes are aligned with phoneme groups which are corresponding to sub-words. For a sub-word,  $w_0$ , suppose it corresponds to phoneme group  $\{PH_{j_0}, PH_{j_1}, PH_{j_2}\}$ . We define  $\tilde{P}_{i,w_0} = \sum_{k=0}^2 P_{i,j_k}$ . Here,  $\tilde{P}$  is the transition probability matrix between phonemes and sub-words (also phoneme groups). Then  $-\tilde{P}$  is the cost matrix used in the DTW. As long as the predicted  $i$ -th frame phoneme is in the phoneme group corresponding to  $w_0$ ,  $\tilde{P}_{i,w_0}$  will be larger than those phoneme groups without it. Therefore, DTW is encouraged to pair the predicted phoneme with the phoneme group including it. For all the pairs

obtained from DTW, if the corresponded phoneme group does not include the predicted phoneme, we always take the phoneme PH with the largest probability in the phoneme group and correct the predicted phoneme to be PH.

## 4. Experiments

The multilingual training dataset is composed of English (en) from LibriSpeech dataset, Korean (kr) from in-house collected dataset, and Kyrgyz (ky), Russian (ru), Swedish (sv), Tatar (tt) from CommonVoice [23] dataset. In the multilingual experiments, the phoneme annotation follows IPA standard. The IPA symbols include 107 letters which represent consonants and vowels, 31 diacritics (or accents), and 17 additional signs which indicate length, tone, stress and intonation. We only take into account consonants and vowels, *i.e.*, the 107 letters, and map other symbols with diacritics and signs to the corresponding IPA letter. We harness MFA to annotate the phonemes which are then mapped into 107 phonemes. For monolingual TIMIT dataset, we use the original 61 annotated phonemes or 39 CMU phonemes according to different experiment settings. For the model training hyperparameters, we set  $\lambda = 4, \epsilon = 0.5$  in training objective loss function, and used Adam optimizer with triangular cyclical learning rate (lr) policy [24] with  $lr_{initial}$  as  $2e-5$ ,  $lr_{max}$  as  $5e-4$  and  $5k$  warm-up steps.

### 4.1. Phoneme Recognition

#### 4.1.1. Comparison with on-device Real-Time SOTA methods

In this experiment, we compare with three other on-device Real-time phoneme recognition models, namely RealPRNet [13], 4 layers LSTM and CLDNN [25]. Here, the training and test data are from TIMIT dataset, which incorporates 61 phoneme symbols. The en PER result is demonstrated in Table 1. We can see that VoiceNet outperforms other on-device models by a large margin.

Table 1: Comparison with on-device Real-Time SOTA methods

	LSTM	CLDNN	RealPRNet	VoiceNet
en PER	18.63%	18.30%	17.20%	10.90%

#### 4.1.2. Comparison with Non-RealTime Multilingual model

In this experiment, we compare VoiceNet with Conformer architecture [3] which has shown competitive performance in ASR. Each Conformer block consists of three modules: a feed-forward module, a multi-head self-attention module, and a convolution module. We followed the small size setting of Conformer (11M parameters) which includes 16 conformer blocks, each with 4 attention heads and the dimension is 144. It is trained with CTC loss. The multilingual training dataset is composed of Librispeech-English (100h), In-house-Korean-training data (50h), CommonVoice-Kyrgyz (3.6h), CommonVoice-Russian (24h), CommonVoice-Swedish (3.1h), CommonVoice-Tatar (20h). The test dataset is composed of Librispeech-test-clean (5.4h), In-house-Korean-test (2.5h), 1h CommonVoice for each of other languages. We convert VoiceNet and Conformer into on-device TFLite model with float16 quantization using Google TensorFlow-Lite Library [26] (currently known as LiteRT). The inference latency is measured on Galaxy S24 mobile phone with 0.5s duration audio as input to the models. When text is available for VoiceNet, DTW is implemented in Java on android and is applied sentence by sentence.

Table 2: Multilingual Phoneme Error Rate results Comparison with on-device SOTA methods.

	en	kr	ky	ru	sv	tt	Avg	# Params.	Latency
Conformer-small	16.4%	20.8%	44.8%	28.9%	45.7%	56.8%	36.3%	11M	90ms
VoiceNet	14.6%	18.5%	42.4%	28.8%	46.0%	54.1%	34.8%	6.4M	6ms
VoiceNet + text	12.2%	17.6%	40.0%	28.2%	45.2%	51.7%	32.5%	6.4M	6.2ms

The test results on en, kr and others (languages of ky, nl, ru, sv and tt) are listed in Table 2. We can see that VoiceNet is able to outperform Conformer by 1.8% in en, 2.3% in kr and 1.5% on average. With text available, VoiceNet could be further improved by 2.3% on average. Also Conformer model consumes more computational resources and incurs much higher inference latency than VoiceNet (6 ms). The high latency of Conformer is partly caused by the TensorFlow Ops which are not built-in TFLite ops and also not supported by mobile GPU and NPU unless with further ops optimization. By contrast, VoiceNet’s architecture contains highly optimized operators for low-latency inference on mobile CPU, GPU and NPU. Although the time complexity of DTW algorithm is  $O(n^2)$ , we apply DTW on sentence level ( $n$  is small), which on average costs 0.2ms for one sentence alignment. Testing on Galaxy S21-S24 mobile phones, one-time inference of VoiceNet could achieve 3ms (2x speedup) on GPU and 0.6ms (10x speedup) on NPU.

## 4.2. Forced Alignment

We compared our method with three publicly available forced aligners: Montreal Forced Aligner (MFA) [27], Penn Forced Aligner (FAVE) [28] and Gentle [29]. We evaluated the forced alignment results with precision (P), recall (R), F1, and R-value [30, 31]. For each predicted phone boundary, if the timing was within tolerance  $\tau$  and the predicted phone matches, it was considered a hit. As each boundary marked the onset and the offset of consecutive phones, we only evaluated the phone onsets with a tolerance of  $\tau=20$ ms. In addition, we also measured the percentage of correctly predicted frame labels at a 10ms time scale. In the experiments, we followed the experiment data settings used in [31]. Since there are very few validated human annotated multilingual test datasets in the Forced-Alignment tasks, we only used English dataset (LibriSpeech-960h) annotated with CMU phonemes as training data for fair comparison with existing phoneme aligners. All the methods were evaluated on TIMIT test dataset with human annotations. The original TIMIT 61 phoneme symbols were collapsed into the 39 CMU phoneme set. In text-dependent experiment, an open-source G2P [32] was used to convert all the transcriptions into phoneme sequences.

### 4.2.1. Text-dependent Alignment

We directly performed MFA, FAVE and Gentle from scratch on the TIMIT test data. W2V2-FC and VoiceNet were trained on LibriSpeech. Both W2V2-FC and VoiceNet involved text in inference as post-processing alignment. Table 3 shows the competitive performance of VoiceNet compared with public aligners and recent work W2V2-FC [31]. Moreover, VoiceNet (6.4M parameters) has much fewer parameters than W2V2-FC (95M parameters) and other public aligners without relying on extra G2P model and targets on-device alignment application.

### 4.2.2. Text-independent Alignment

Table 4 showed that the alignments generated by W2V2-CTC method are significant worse than other methods since CTC loss alone does not encourage time-synchronous alignment. W2V2-FC outperforms VoiceNet by a margin while the number of pa-

Table 3: Comparison with text-dependent alignment methods

Model	P	R	$F_1$	R-val	Overlap
MFA	0.62	0.63	0.63	0.68	75.0%
FAVE	0.57	0.59	0.58	0.64	74.3%
Gentle	0.49	0.46	0.48	0.56	67.7%
W2V2-FC + text	0.57	0.54	0.55	0.62	76.4%
VoiceNet + text	0.55	0.54	0.54	0.62	72.5%

rameters of W2V2-FC is magnitudes of VoiceNet. We could also observe that the extra transcript enhanced VoiceNet  $F_1$  and R-val by around 2%, also with a 4.1% improvement for overlapping accuracy, which shows that our method is effective for phoneme recognition.

Table 4: Comparison with text-independent alignment methods

Model	P	R	$F_1$	R-val	Overlap	# Params.
W2V2-CTC	0.31	0.29	0.30	0.42	43.9%	95M
W2V2-FC	0.55	0.58	0.56	0.62	72.5%	95M
VoiceNet	0.53	0.51	0.52	0.60	68.4%	6.4M

## 4.3. Ablation Study

In this experiment, we explored the contribution of different components to VoiceNet. We proposed several variants of VoiceNet, (1) VoiceNet.1: a shallow VoiceNet variant which only repeats Block 1 by seven times without self-attention layer and SE layers; (2) VoiceNet.2: VoiceNet variant without self-attention layer; (3) VoiceNet.3: VoiceNet variant without SE layers; (4) VoiceNet: the final VoiceNet model. The training data is composed of LibriSpeech-960h and In-house-Korean-50h datasets, and the test data is composed of LibriSpeech-test-clean and In-house-Korean-test dataset. Via phoneme-viseme mapping according to [33], we could obtain corresponding visemes, which animate the lips to sync with given audio. Each viseme corresponding to multiple phonemes, with the lower Viseme Error Rate (VER), we expect more accurate lip-Sync. The evaluation metrics of PER and VER are listed in the following Table 5, from which we could observe that self-attention layer and SE layers are important to VoiceNet model.

Table 5: Ablation Study of different variants of VoiceNet.

	en PER	en VER	kr PER	kr VER
VoiceNet.1	17.9%	12.8%	24.1%	18.3%
VoiceNet.2	14.4%	10.5%	18.4%	14.1%
VoiceNet.3	13.4%	9.9%	17.6%	13.5%
VoiceNet	11.2%	8.3%	15.2%	11.8%

## 5. Discussion and Conclusion

In this work, we proposed Voicenet, a real-time on-device computationally efficient end-to-end multi-lingual phoneme recognition and text-independent forced alignment model. To our best knowledge, this is the first on-device text-dependent phoneme-to-audio alignment without extra G2P model for lip-sync. In the streaming application of digital assistant enabled by LLM and TTS, our VoiceNet model will play an increasingly important role on mobile phones. In the future, we will continue to improve the alignment accuracy and reduce the inference latency, so that it can be applied in even resource-limited devices, for example, Smart-Watch and Smart-Glasses.

## 6. References

- [1] P. Edwards, C. Landreth, E. Fiume, and K. Singh, “Jali: an animator-centric viseme model for expressive lip synchronization,” *ACM Transactions on graphics (TOG)*, vol. 35, no. 4, pp. 1–11, 2016.
- [2] N. Christoff, N. N. Neshov, K. Tonchev, and A. Manolova, “Application of a 3d talking head as part of telecommunication ar, vr, mr system: Systematic review,” *Electronics*, vol. 12, no. 23, p. 4788, 2023.
- [3] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” *Interspeech 2020*, 2020.
- [4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
- [5] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, “Contextnet: Improving convolutional neural networks for automatic speech recognition with global context,” *arXiv preprint arXiv:2005.03191*, 2020.
- [6] S. Majumdar, J. Balam, O. Hrinchuk, V. Lavrukhin, V. Noroozi, and B. Ginsburg, “CitriNet: Closing the gap between non-autoregressive and autoregressive end-to-end models for automatic speech recognition,” *arXiv preprint arXiv:2104.01721*, 2021.
- [7] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, “Jasper: An end-to-end convolutional neural acoustic model,” *arXiv preprint arXiv:1904.03288*, 2019.
- [8] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, “Rnn-transducer with stateless prediction network,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7049–7053.
- [9] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, “A comparison of sequence-to-sequence models for speech recognition,” in *Interspeech*, 2017, pp. 939–943.
- [10] A. Abe, K. Yamamoto, and S. Nakagawa, “Robust speech recognition using dnn-hmm acoustic model combining noise-aware training with spectral subtraction,” in *Interspeech*, 2015, pp. 2849–2853.
- [11] J. Park, S. Jin, J. Park, S. Kim, D. Sandhyana, C. Lee, M. Han, J. Lee, S. Jung, C. Han *et al.*, “Conformer-based on-device streaming speech recognition with kd compression and two-pass architecture,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 92–99.
- [12] M. Xu, A. Jin, S. Wang, M. Su, T. Ng, H. Mason, S. Han, Z. Lei, Y. Deng, Z. Huang *et al.*, “Conformer-based speech recognition on extreme edge-computing devices,” *arXiv preprint arXiv:2312.10359*, 2023.
- [13] Z. Yu, H. Wang, and J. Ren, “Realprnet: A real-time phoneme-recognized network for “believable” speech animation,” *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5357–5367, 2021.
- [14] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, 2021.
- [15] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. Von Platen, Y. Saraf, J. Pino *et al.*, “Xls-r: Self-supervised cross-lingual speech representation learning at scale,” *arXiv preprint arXiv:2111.09296*, 2021.
- [16] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.
- [17] “International phonetic alphabet,” [https://en.wikipedia.org/wiki/International\\_Phonetic\\_Alphabet](https://en.wikipedia.org/wiki/International_Phonetic_Alphabet), accessed: 2025-02-12.
- [18] Z. Leng, M. Tan, C. Liu, E. D. Cubuk, X. Shi, S. Cheng, and D. Anguelov, “Polyloss: A polynomial expansion perspective of classification loss functions,” *arXiv preprint arXiv:2204.12511*, 2022.
- [19] X. Yang, A. Loukina, and K. Evanini, “Machine learning approaches to improving pronunciation error detection on an imbalanced corpus,” in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 300–305.
- [20] J. Zhu, C. Zhang, and D. Jurgens, “Byt5 model for massively multilingual grapheme-to-phoneme conversion,” *arXiv preprint arXiv:2204.03067*, 2022.
- [21] C. Wang, P. Huang, Y. Zou, H. Zhang, S. Liu, X. Yin, and Z. Ma, “Liteg2p: A fast, light and high accuracy model for grapheme-to-phoneme conversion,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [22] H. Xu, S. Ding, and S. Watanabe, “Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7110–7114.
- [23] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, May 2020, pp. 4218–4222. [Online]. Available: <https://aclanthology.org/2020.lrec-1.520/>
- [24] L. N. Smith, “Cyclical learning rates for training neural networks,” in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.
- [25] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. Ieee, 2015, pp. 4580–4584.
- [26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, and e. a. Greg S. Corrado, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [27] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using kaldii,” in *Interspeech*, vol. 2017, 2017, pp. 498–502.
- [28] I. R. *et al.*, “Forced alignment and vowel extraction (fave) program suite,” 2011.
- [29] “Robust yet lenient forced-aligner built on kaldii. a tool for aligning speech with text.” <https://github.com/strob/gentle>.
- [30] F. Kreuk, J. Keshet, and Y. Adi, “Self-supervised contrastive learning for unsupervised phoneme segmentation,” *arXiv preprint arXiv:2007.13465*, 2020.
- [31] J. Zhu, C. Zhang, and D. Jurgens, “Phone-to-audio alignment without text: A semi-supervised approach,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8167–8171.
- [32] K. Park and J. Kim, “g2pe,” <https://github.com/Kyubyong/g2p>, 2019.
- [33] Y. Zhou, Z. Xu, C. Landreth, E. Kalogerakis, S. Maji, and K. Singh, “Visemenet: Audio-driven animator-centric speech animation,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–10, 2018.