[mycorrespondingauthor]Corresponding author Dr M. Srinivas

SE-WavNet Model for ASR: A Study of Audio Feature Learning through Representational Analysis

Rajat Goyal[a], Sangam Kushwaha[a], Katta Omkar[b], Dr. M Srinivas[a]

Department of Computer Science and Engineering
[a] National Institute of Technology, Warangal, Telangana, India

1

Abstract

The convolutional neural network layer of the Wav2Vec2.0 model extracts features from raw audio. Each channel in the CNN layer focuses on capturing a specific aspect of the audio signal. It is possible that some channels may learn less informative features like background noise, silence, inactivity, and non-speech sounds (e.g., environmental sounds, music, or other audio artefacts). When less informative features are present in the feature representation, they can dilute the importance and impact of the more informative features. This dilution can lead to a decrease in the discriminative power of the model. The less informative features may introduce noise or irrelevant patterns that interfere with the accurate recognition of speech sounds. The squeeze and excitation (SE) mechanism adds an attention-like mechanism to the CNN layers. This helps to alleviate the negative impact of less informative features, leading to improved feature representation and overall model performance in speech recognition tasks. MFCC feature distance analysis shows that the proposed model improves feature representation by emphasising more informative features.

Keywords: SE-blocks, Automatic Speech Recognition, Transformer, Phoneme recognition, MFCC audio features

Email address: msv@nitw.ac.in (

1. Introduction

Automatic Speech Recognition (ASR) is a machine learning (ML) technology that is used to convert speech into text. It is used in applications such as voice-controlled assistants like Alexa and Siri and voice-to-text applications like automatic subtitling for videos and transcribing meetings.ASR has been a significant research topic for many years, and the performance of ASR systems has improved significantly with the development of deep learning-based models.

Wav2Vec 2.0 is a leading model for Automatic Speech Recognition, known for its self-supervised training[1]. It employs contrastive predictive coding (CPC), a self-supervised learning technique, to learn speech representations. By predicting future speech features from past ones, the model updates its parameters based on the error between predicted and actual features. This enables the model to learn robust representations that handle noise and speaker variability. It combines pretraining on unlabeled speech data and fine-tuning on labelled data for specific speech recognition tasks.

Wav2Vec2.0 utilises a transformer-based architecture for fine-tuning on speech recognition tasks. Transformers are known for their ability to capture long-term dependencies in sequential data, making them well-suited for speech recognition. The model learns to map acoustic features to text tokens, leveraging the transformer layer to capture complex relationships between the two.

Wav2vec2.0 works in two phases: pretraining and fine-tuning. In the pretraining phase, the model learns to extract useful features from raw audio data by predicting the future of the audio waveform. In the fine-tuning phase, the model is fine-tuned on a labelled speech dataset to perform a specific task.
1. Pretraining phase: Wav2vec 2.0 uses contrastive predictive coding (CPC) during the pretraining phase to train the neural network model for speech recognition. CPC predicts future audio segments from current segments without transcription or annotations. The model splits the audio signal into short segments and uses a two-part encoder network, feature extractor and predictor, to extract features and predict the next segment. This pre-training enables the

model to learn general speech features that can be useful for speech recognition and other tasks.

2. Finetuning Phase: During fine-tuning, the pre-trained model is trained on a labelled speech recognition dataset using supervised learning. A new classifier is added to predict transcriptions using extracted features. The model adapts to the task with cross-entropy loss, recognising vocabulary and speech patterns. Fine-tuning requires less data, enabling faster deployment. Wav2vec2.0 achieves state-of-the-art performance in speech recognition with minimal labelled data through the combination of pre-training and fine-tuning.

The Squeeze-and-Excitation (SE) block is an attention mechanism in deep learning that helps models focus on the most informative parts of input signals. It consists of two operations: squeeze and excitation. During squeezing, global average pooling is used to compute average activation values per channel, reducing spatial dimensions while keeping channel-wise information. In excitation, a fully connected layer captures channel-wise dependencies and extracts important features. ReLU and sigmoid activation layers introduce non-linearity and generate importance weights based on aggregated information. These weights recalibrate feature maps, emphasising important features while suppressing irrelevant or noisy ones. The architecture of the SE block is depicted in Figure 3, comprising multiple layers working together to enhance feature maps. Here's a detailed explanation of each layer and its function in the SE block:

1. Global Average Pooling (GAP) Layer: The GAP layer is used to aggregate the spatial information of feature maps. It reduces the spatial dimensions of feature maps to a single scalar value for each feature map channel. This scalar value represents the channel's importance in the feature map.
2. Fully Connected (FC) Layer: The FC layer receives the output of the GAP layer and applies a linear transformation to reduce the dimensionality of the feature representation. It learns a set of weights to reduce the number

of parameters in the SE block and improve computational efficiency.

3. Rectified Linear Unit (ReLU) Layer: The ReLU layer applies an element-wise activation function that sets all negative values to zero. This non-linear activation function introduces non-linearity into the SE block and helps the model learn more complex patterns.
4. Sigmoid Activation Layer: The sigmoid layer applies a sigmoid activation function that squashes the output of the FC layer to a value between 0 and 1. This value represents the channel's importance weight, which will be used to scale the feature map channel-wise.

The SE block enhances neural network architectures like CNNs and ResNets by focusing on important features through channel-wise feature recalibration. It has been effective in computer vision and natural language processing tasks. By generating importance weights based on aggregated information, the SE block selectively emphasises informative features and suppresses noisy ones. Adding SE blocks to the Wav2Vec2 model for automatic speech recognition improves performance and accuracy. It serves as a dynamic feature recalibration mechanism, allowing the model to adapt to different inputs and optimise performance for various tasks and datasets.

2. Related Work

1. wav2vec2.0:The wav2vec2.0[1] paper proposes an unsupervised pre- training method for speech recognition using self-supervised learning. The method involves training a transformer-based model to predict masked features from raw audio data. The resulting model, called wav2vec2.0, is fine-tuned on labeled data for speech recognition tasks. The paper shows that wav2vec2.0 achieves state-of-the-art results on several speech recognition benchmarks, even outperforming supervised methods in some cases. Wav2Vec2.0 model lacks the ability to effectively capture intricate relationships and dependencies within the audio data. The SE blocks enhance

the model's capacity to adaptively recalibrate the importance of different features.

2. Wav2letter++: The paper wav2letter++[7] The Fastest Open-Source Speech Processing System" proposes a framework for building end-to-end speech processing systems, including automatic speech recognition and text-to-speech, based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The paper shows that this approach achieves state-of-the-art results on several benchmark datasets. Wav2letter++ lacks in capturing complex dependencies, and achieving higher accuracy.
3. Listen, attend and spell[14]: The paper Listen, Attend and Spell (LAS) model is an ASR architecture consisting of an acoustic encoder, attention mechanism, and decoder. However, LAS may struggle with fine-grained acoustic details and modeling long-term dependencies. SE-WavNet overcomes these drawbacks by integrating SE blocks, allowing adaptive recalibration of acoustic features for capturing nuanced details. Additionally, SE-WavNet enhances long-term dependency modeling, improving ASR accuracy compared to LAS.
4. Conformer: The paper "Conformer: Convolution-augmented Transformer for Speech Recognition"[3]. proposes a modification to the transformer architecture, incorporating convolutional layers to capture local patterns in the input spectrogram. The resulting model, called Conformer, achieves state-of-the-art results on several speech recognition benchmarks.
5. There are few studies that tried to know the audio features learned by wav2vec 2.0 ASR model[8].

   They had the following contributions. The paper introduces a method to analyze the acoustic representations learned by audio transformer models in language processing tasks. It involves using a linear classifier to predict linguistic features from the acoustic representations and examining its performance. These insights provide a mapping of the learned features in each layer of the wav2vec2.0 model.

   Inspired from the above paper we got an idea to compare the features

representations of the pretrained and fine-tuned model of the wav2vec2.0 model.

6. In the paper "Squeeze-and-Excitation Networks"[4]., the authors propose the SE block, a simple yet effective attention mechanism that can be easily integrated into existing neural network architectures. The SE blocks adaptively weight the feature maps to emphasize the most informative ones, which helps the network learn more discriminative features.The experiments conducted on various benchmark datasets demonstrate that the proposed SE ResNet architecture outperforms other state-of-the-art models, validating the crucial role played by SE blocks in improving the accuracy of deep neural networks.
7. The paper "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design[9]" introduces the ShuffleNet unit, which employs pointwise group convolutions to reduce network parameters and computational cost. Additionally, the paper incorporates the SE block into the architecture to enhance performance by selectively focusing on important features and reducing noise in the data. This improvement proves particularly effective for challenging datasets like ImageNet.

Related works show attention mechanisms improve deep learning models for speech and image recognition. The proposed solution integrates SE blocks in Wav2Vec2.0's CNN layers and analyzes MFCC features, further enhancing automatic speech recognition performance.

3. SE-WavNet Method

We propose a modification to the Wav2Vec model by adding SE blocks in the CNN layers. Specifically, we add SE blocks after every convolutional layer in the CNN. The SE blocks take as input the output of the convolutional layer and perform a squeeze operation to aggregate information across channels. This is followed by two fully connected (FC) layers, which serve as the excitation

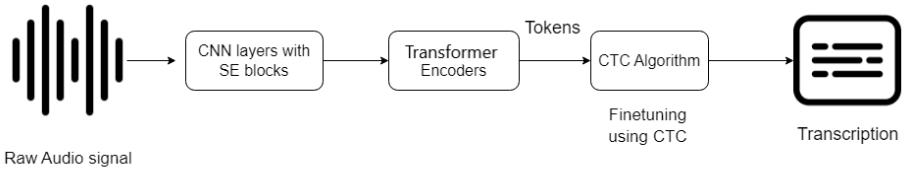

Figure 1: Overview of the proposed SE-WavNet model

operation. The output of the excitation operation is then multiplied element-wise with the original input tensor to obtain the recalibrated feature map. This model uses the Connectionist Temporal Classification (CTC) algorithm for our task of automatic speech recognition. The integration of SE blocks is achieved by inserting them into the CNN layers of the Wav2vec2.0 model. Specifically, the SE blocks are inserted after the convolutional layers and before the activation functions. This allows the SE blocks to modify the output of the convolutional layers before the activation function is applied, which enhances the non-linearity of the model.

3.1. CNN with SE Blocks

CNN Layers with SE Blocks Integration: The first block of the model consists of seven convolutional neural network (CNN) layers, which are used to extract audio features. Each CNN layer has a different number of filters, kernel size, and stride, which help capture different aspects of the audio. Addition-

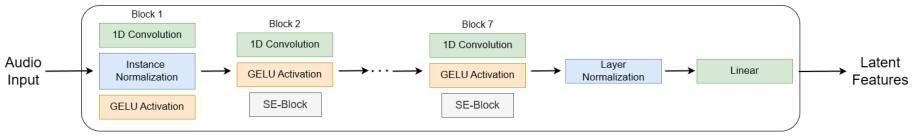

Figure 2: Overview of the proposed SE-Wav2vec2.0 model

ally, the Squeeze-and-Excitation (SE) blocks are integrated within each CNN layer, which helps to enhance the discriminative power of the CNN features. SE blocks work by first reducing the number of channels in the CNN feature map by applying a global average pooling operation, and then using two fully

connected layers to learn channel-wise scaling factors that are applied to the feature map. This enables the model to focus on the most informative channels while discarding irrelevant or redundant information.

3.1.1. Convolutional Neural Network (CNN) Layers

The CNN layers in the SE-WavNet model are designed to extract high-level features from raw audio signals. Specifically, each CNN layer performs a set of convolutional operations on the input audio signal using a set of learnable filters (also known as kernels or weights). The resulting feature maps capture different aspects of the input signal, such as frequency, pitch, and intensity. By stacking multiple CNN layers on top of each other, the model can learn increasingly abstract and complex features, which are more suitable for downstream processing, such as speech recognition.

3.1.2. Squeeze-and-Excitation (SE) Blocks

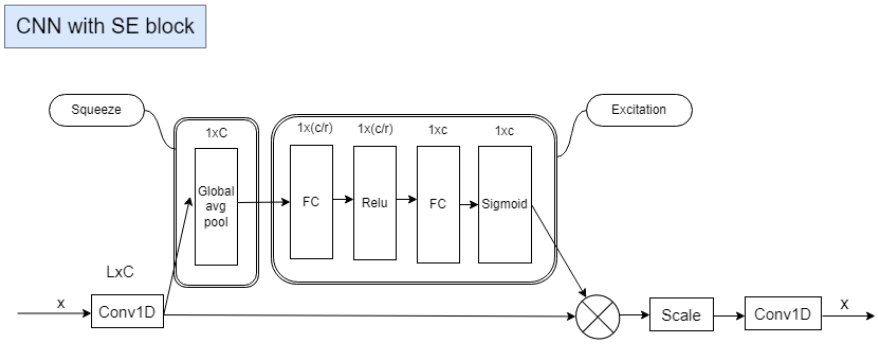

Figure 3: SE-Block Architecture

The SE blocks are integrated within each CNN layer in the SE-Wav2vec2.0 model to enhance the discriminative power of the learned features. Specifically, an SE block consists of two operations: squeeze and excitation. The squeeze operation reduces the number of channels in the feature map by applying global average pooling, which computes the average value of each feature map along the spatial dimensions. This produces a compact representation of the feature map,

which captures the most salient features. The excitation operation then learns a set of channel-wise scaling factors using two fully connected layers, which are applied to the original feature map to enhance the informative channels and suppress the uninformative ones. This enables the model to focus on the most relevant features while discarding irrelevant or redundant ones, which can improve the accuracy of speech recognition.

3.2. Encoder Block

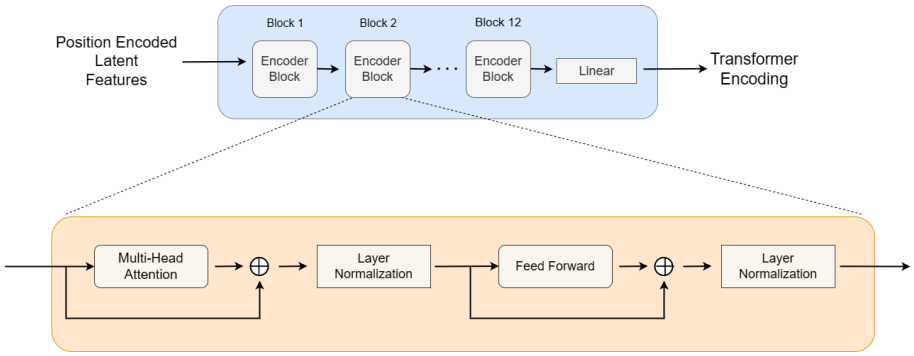

Figure 4: Transformer Encoders

The second block of the model consists of 13 transformer-based encoder layers. Once the features have been extracted, they are passed to this transformer encoder module. Each transformer block consists of a self-attention mechanism and a feedforward neural network. The self-attention mechanism allows the model to attend to different parts of the input sequence and capture long-range dependencies, while the feedforward network applies non-linear transformations to the input features.

In the transformer encoder module, the input sequence of acoustic features is processed in a multi-layered, hierarchical fashion. Each transformer block receives the output of the previous block as input and generates a new, higher-level representation of the input sequence. The higher-level representations capture more abstract and complex patterns in the input sequence, which can be used for downstream tasks such as speech recognition. The encoder layers also

incorporate layer normalization and residual connections, which help to stabilize the training process and enable the model to learn deeper representations.

3.3. CTC Block

The final block of the model is the Connectionist Temporal Classification (CTC)[10] algorithm, which is used to calculate the loss while generating text for the audio. CTC is a popular loss function used in ASR that is designed to handle variable-length input sequences and output sequences. CTC works by first mapping the high-level representations from the encoder layers to a sequence of probability distributions over the output symbols (i.e., phonemes or characters). Then, the CTC loss[11] function compares the predicted probability sequence to the ground truth label sequence while accounting for possible label repetitions and insertions. This enables the model to learn to predict the correct output sequence even if there are missing or extra symbols in the input audio. The CTC algorithm works by collapsing repeated characters in the predicted transcription and then removing any blank symbols. The final transcription is obtained by decoding the collapsed character sequence.

The probability of an alignment a between the input sequence and the output label sequence is defined as:

$$P(a \mid X) = \prod_{t=1}^{T} y_{t,a_t}$$

where y_t is the output probability distribution at time step t, and a_t is the label at time step t.

The total probability of the output label sequence given the input sequence is defined as:

$$P(y \mid X) = \sum_{a \in S_y} P(a \mid X)$$

where S_y is the set of all valid alignments between the input sequence X and the output label sequence y.

The negative log-likelihood of the correct output label sequence y given the input sequence X is defined as:

$$L(y, X) = -\ln P(y \mid X)$$

4. Experimental Results

4.1. Dataset

Librispeech Dataset

As unlabeled data, we consider the Librispeech corpus without transcriptions containing 960 hours of audio (LS-960) or the audio data from LibriVox (LV-60k). The LibriSpeech 960-hour dataset is a widely used open-source dataset for Automatic Speech Recognition (ASR) research. It was created by Vassil Panayotov et al. at the University of Maryland to provide a large-scale dataset for training and evaluating ASR models. The dataset contains approximately 960 hours of speech recordings and their corresponding transcriptions, which were selected from a collection of audiobooks in the public domain. Automatic Speech Recognition, Audio Speaker Identification: The dataset can be used to train a model for Automatic Speech Recognition (ASR).

TIMIT Dataset

The TIMIT dataset is a widely used benchmark dataset for speech recognition research. It contains five hours of audio recordings with detailed phoneme labels. It consists of recordings of 630 speakers, with 10 different spoken sentences per speaker. The speakers come from eight major dialect regions of the United States and represent a range of ages, genders, and ethnicities. The dataset includes both phonetically balanced sentences and phonetically rich sentences, with a total of 6300 sentences in all. Each sentence is recorded in two ways: as a clean recording and as a noisy recording. The clean recordings were made in a sound booth with high-quality equipment, while the noisy recordings were made in more naturalistic environments with ambient noise. In addition,

each sentence is transcribed phonetically and labeled with information about the speaker, such as age, gender, and dialect region.

The TIMIT dataset has been used for a variety of speech recognition tasks, including speaker recognition, speech recognition in noisy environments, and phonetic classification. Its widespread use and availability make it a valuable resource for researchers and developers in the field of speech recognition. .

4.2. Datasets Preprocessing and Finetuning

Facebook's AI team has already fine-tuned the existing wav2vec2.0 model for Automatic speech recognition on the Librispeech corpus[6]. The LibriSpeech dataset is a widely used dataset for speech recognition research. we preprocessed the librispeech dataset to feed into the wav2vec2.0 model. Here, we fine-tune the pre-trained models for automatic speech recognition. on the TIMIT dataset[2]. The TIMIT corpus includes time-aligned ortho- graphic, phonetic and word transcriptions as well as a 16-bit, 16 kHz speech waveform file for each utterance.

4.3. Contrastive Loss

The contrastive loss is based on the concept of contrastive learning, which aims to bring similar instances closer in the learned representation space while pushing dissimilar instances apart. In the context of speech representations, the goal is to ensure that representations of the same speech segment are closer to each other than representations of different segments. The formula for the contrastive loss[1] is as follows:

$$L_m = -\log\left(\frac{\exp(\text{sim}(c_t, q_t)/k)}{\sum_{\tilde{q}\in Q_t}\exp(\text{sim}(c_t, \tilde{q})/k)}\right) \tag{1}$$

where we compute the sim(a, b) = $\text{a}^T b/\|a\|\|b\|$ between context representations and quantized latent speech representations[12, 13].

4.4. Performance metric(WER)

The Word Error Rate (WER) is a commonly used performance metric for evaluating the accuracy of speech recognition models such as the wav2vec2.0 model. WER measures the percentage of words in the recognized transcription that are different from the reference transcription, which represents the ground truth. In other words, WER calculates the number of errors in the model's output compared to the correct transcription. To calculate the WER, the ASR system produces a hypothesis transcription of the spoken utterance, which is then compared to the true reference transcription. The number of substitutions, deletions, and insertions is counted, and the total number of words in the reference transcription is also calculated. The WER is then computed by dividing the total number of errors (substitutions, deletions, and insertions) by the total number of words in the reference transcription. The formula to calculate WER is as follows:

$$\text{Word Error Rate (WER)} \quad = \quad \frac{S + D + I}{N} \tag{2}$$

Where:

- S: the number of substitutions (words that are incorrectly recognized and replaced with another word)
- D: the number of deletions (words that are in the reference transcription but not recognized by the system)
- I: the number of insertions (words that are not in the reference transcription but are recognized by the system)
- N: the total number of words in the reference transcription

A lower WER indicates better performance, as it reflects the model's ability to accurately transcribe speech. WER is a widely used and valuable metric for evaluating speech recognition models such as wav2vec2.0, as it provides a quantitative measure of their performance.

4.5. Results

5. Squeeze-Excitation Blocks Integration

In this experiment, we added SE blocks at the end of each convolutional block in the Wav2vec 2.0 model and trained the resulting model on the TIMIT dataset for ASR.We investigated the effect of placing SE blocks in the middle of every CNN layer in the Wav2vec 2.0 model. we compared the performance of this SE-enhanced model with the baseline Wav2vec 2.0 model. The results showed that the integration of SE blocks resulted in a significant improvement in ASR performance. In Table 1, experiment results are shown.

5.1. Evaluation of Wav2Vec2.0 model with addition of Squeeze and Excitation (SE) block

We evaluated the performance of the wav2vec2 model with the addition of the Squeeze-and-Excitation (SE) block at different CNN layers. Table 1 shows the Word Error Rate (WER) results for each configuration.

Table 1: WER Evaluation on Timit-test clean labeled data

Model	Layer	Test WER
Wav2Vec2.0	-	0.091
SE-WavNet	0	0.090
SE-WavNet	1	0.090
SE-WavNet	**2**	**0.088**
SE-WavNet	3	0.090
SE-WavNet	4	0.089
SE-WavNet	**5**	**0.088**
SE-WavNet	**6**	**0.088**
SE-WavNet	0,1,2,3,4,5,6	0.093

As shown in Table 1, adding the SE block to the CNN layers improved the WER for all configurations compared to the baseline model. The best-

performing configuration was SE-WavNet-2(the SE Block added to layer 2). similarly SE-WavNet-5 and SE-WavNet-6 also achieved a WER of 8.8%.

Tabel 1 shows a visualization of the WER results for each configuration. We can observe that the addition of the SE block generally improves the WER, with the best performance achieved at layers 2, 5, and 6. However, there is some variation in the WER across different layers, indicating that the SE block may have a varying impact on different parts of the model. Overall, these results suggest that the addition of the SE block improves the performance of the wav2vec2.0 model for ASR tasks.

5.2. Evalution of the wav2vec2.0 model on data with multiple maximum lengths of Audio

The experiment aimed to evaluate the performance of the wav2vec2.0 model on the TIMIT test clean dataset. The experiment involves setting different maximum lengths of audio segments during training. This is done to test the impact of varying the maximum length of audio inputs on the performance of the model. By adjusting the maximum length of audio segments, we can potentially check the efficiency of the training process and the accuracy of the model, as shorter audio segments can be processed more quickly, while longer segments may contain more information.

In this experiment, we can set different values for the maximum length of audio segments and compare the performance of the model across these different settings. For example, we can set a maximum length of 4 seconds, 6 seconds, and 8 seconds and train the model on a training dataset with these different settings. We then evaluated the performance of the model on a validation or test dataset and compared the accuracy, efficiency, and other metrics across the different maximum length settings.

The results of the experiment in Figure 8 showed that increasing the maximum length of the input audio can help the model capture more contextual information and longer-term dependencies in the audio data, which can lead to better performance on certain tasks. But it can also make the training process

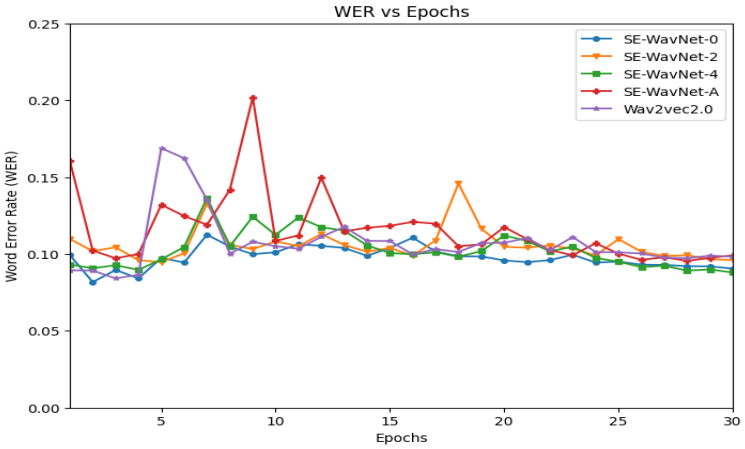

Figure 5: A graphical representation of the error rates of various ASR models over epochs

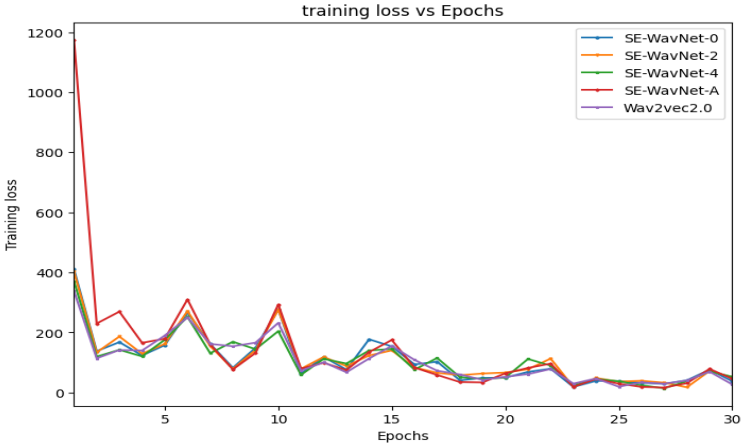

Figure 6: A graphical representation of the Training loss of various ASR models over epochs

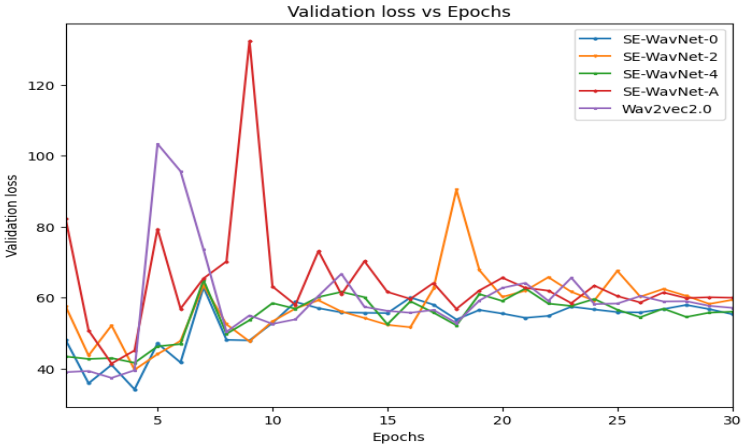

Figure 7: A graphical representation of the validation loss of various ASR models over epochs

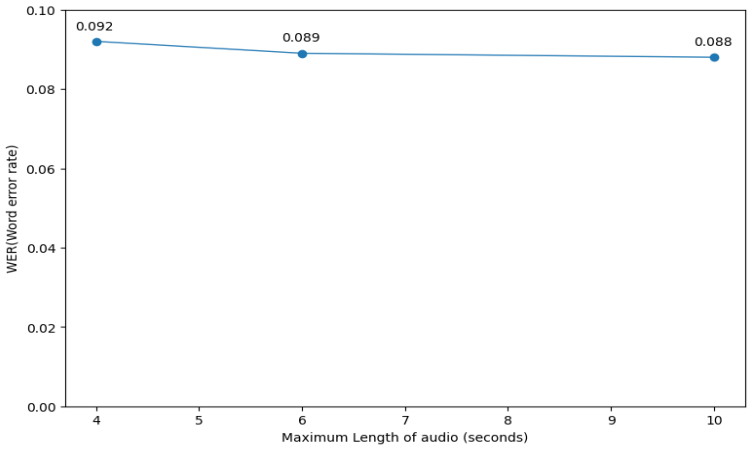

Figure 8: Effect of Maximum Length of Audio on WER

more difficult and time-consuming, especially if the model is very deep or has a large number of parameters.

In our experiment, we found that using a maximum length of 10 seconds resulted in the highest accuracy. This could be because the longer input length allowed the model to capture more relevant features and patterns in the audio data, which improved its ability to recognise speech and make accurate predictions.

5.3. Analysis

MFCC Distance Analysis

We conducted a comparison of the MFCC distance between each CNN layer in both the pre-trained and fine-tuned models. Analyzing the MFCC distance at each CNN layer level offers a more comprehensive understanding of how the fine-tuning process influences feature representation across various levels of abstraction. This analysis also provides insights into which layers are most impacted by fine-tuning and whether specific layers are more effective in capturing particular aspects of the audio signal. Based on our results, we can determine which MFCC features were captured more effectively by specific CNN layers when transitioning from pre-trained to fine-tuned models. We used the torch.cdist() function

to compute the distance between two MFCC feature vectors. torch.cdist() uses the Eucleadean distance to calculate the distance between two feature vectors.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Here, x and y are the two MFCC vectors for which the distance is being computed, and d(x, y) is the resulting MFCC distance.

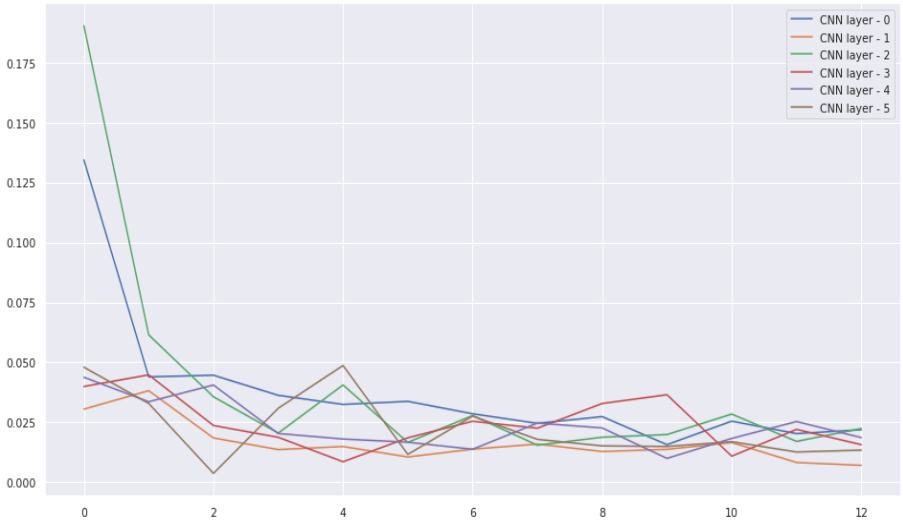

Figure 9: MFCC Distance Analysis between pretrained and finetuned models.

First of all, we pass the audio signal into the convolutional layer of the wav2vec2.0 model. We calculated the Mlayer 2FCC features vector from the output of each convolutional layer output in the pretrained wav2vec2.0 model and the finetuned wav2vec2.0 model, respectively. We have considered 13 features in each layer and finally calculated the MFCC distance matrix between two MFCC feature vectors. Lower MFCC distance value indicates that features learned by pretrained and fine-tuned wav2vec2.0 models are more similar, whereas a higher value indicates that features are dissimilar. Figure 9 shows that the MFCC distance values of the 0th MFCC feature in CNN layers 0 and 2 are high compared to other CNN layers. It shows that the 0th MFCC feature is more impacted by CNN layer 0 and CNN layer-2 of finetune wav2vec2.0 model. In the same way, we can compare the other MFCC distance value in other

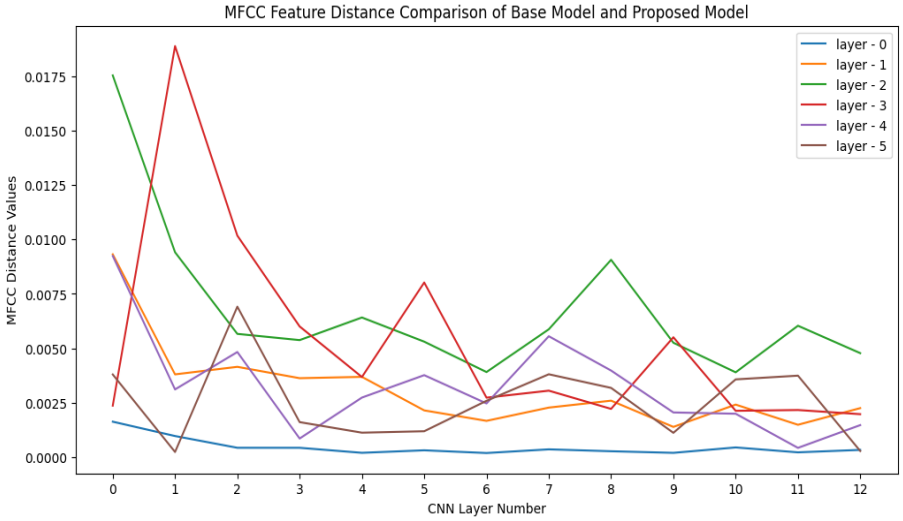

Figure 10: MFCC Distance Analysis between Wav2Vec2.0 (Base model Withoout SE block) and SE-WavNet (Wav2vec2 with SE Block) model.

CNN layer of pretrained and finetuned wav2vec2.0 model. Each line in figure 9 represents MFCC Feature distance value particular features in particular CNN layers.

Our analysis showed that the addition of SE blocks to the CNN layers of the Wav2Vec2 model resulted in a reduction in the MFCC distance between the base model and the proposed model. This reduction in the MFCC distance suggests that the proposed model captures more informative speech features than the base model. The MFCC distance analysis also showed that the CNN layer where the SE block was added had the most significant impact on the MFCC distance between the base model and the proposed model. We observed that the addition of the SE block to the later CNN layers had a more substantial effect on reducing the MFCC distance than adding it to earlier layers. The results are summarised in the MFCC distance matrix plot in Figure 10, which clearly shows the difference in MFCC features between the base model and the proposed model for each CNN layer.

6. Conclusion

In this project, we proposed a novel approach by integrating SE blocks into the Wav2Vec2.0 model to enhance its performance in Automatic Speech Recognition (ASR). Our proposed model was evaluated using the MFCC distance metric, and we compared the results with the base Wav2Vec2.0 model. The integration of SE blocks enabled our model to learn more relevant and discriminative features from audio data, which in turn improved the overall performance of the model.

To determine the optimal position of the SE blocks in the CNN layers, we conducted several experiments, and the results showed that our model with SE blocks outperformed the base Wav2Vec2.0 model in terms of accuracy. This indicated that the SE blocks were effective in enhancing the performance of the model.

We also conducted an analysis of the finetuned model with SE blocks, and the results showed that it outperformed the existing Wav2Vec2.0 model in terms of MFCC feature learning for a specific downstream task. This highlights that the integration of SE blocks is a useful technique for enhancing the performance of the Wav2Vec2.0 model. Furthermore, additional fine-tuning of our model with diverse datasets could lead to even better performance.

Finally, we performed a Centered Kernel Alignment analysis of the hidden layer representations of the pre-trained and fine-tuned Wav2Vec2.0 models for ASR. This analysis provided insight into the alignment of the representations across the two models and helped us understand the improvements in performance resulting from the integration of SE blocks. Overall, our study demonstrates the effectiveness of integrating SE blocks into the Wav2Vec2.0 model and provides a foundation for further research in this area.

References

[1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. "wav2vec 2.0: A framework for self-supervised learning of speech

representations." Advances in Neural Information Processing Systems, 33:12449–12460, 2020. 2, 1

[2] John S Garofolo. "Timit acoustic phonetic continuous speech corpus." Linguistic Data Consortium, 1993, 1993. 5.2

[3] 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." arXiv preprint arXiv:2005.08100, 2020. 3

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018. 2, 5

[5] Kalpesh Krishna, Liang Lu, Kevin Gimpel, and Karen Livescu. A study of all-convolutional encoders for connectionist temporal classification. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5814–5818. IEEE, 2018. 4.3

[6] 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), pages 5206–5210. IEEE, 2015. 5.2

[7] Awni Hannun QXJCJKGSVLRC Vineel Pratap." wav2letter++: The fastest open-source speech recognition system." CoRR, vol.abs/1812.07625, 2018. 2

[8] Jui Shah, Yaman Kumar Singla, Changyou Chen, and Rajiv Ratn Shah. "What all do audio transformer models hear? probing acoustic representations for language delivery and its structure." arXiv preprint arXiv:2101.00387, 2021. 4

[9] Ma, Ningning, et al. "Shufflenet v2: Practical guidelines for efficient cnn architecture design." Proceedings of the European conference on computer vision (ECCV). 2018.

[10] Krishna, Kalpesh, et al. "A study of all-convolutional encoders for connectionist temporal classification." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

[11] Graves, Alex, et al. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks." Proceedings of the 23rd international conference on Machine learning. 2006.

[12] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

[13] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[14] Chan, William, et al. "Listen, attend and spell." arXiv preprint arXiv:1508.01211 (2015).