Ink to Digital: Handwritten Word Recognition

Rajeswari Depala
rdepala@hawk.iit.edu
A20526535

1.ABSTRACT

The study involves incorporating convolutional layers, bidirectional long short-term memories (LSTMs), attention mechanisms, and a CTC (Connectionist Temporal Classification) layer. This method presents a novel approach for addressing sequence-to-sequence tasks. Optical character recognition (OCR) and handwriting word recognition are two typical applications of this technology.

The model is composed of convolutional layers for extracting image features, along with recurrent Bidirectional LSTM layers and attention mechanisms for processing sequences. The last dense layer generates character predictions. The model utilizes a CTC layer for sequence alignment. The IAM dataset consists of handwritten text in a script style. This allows characters within words to appear "overlapping." A thorough analysis of an IAM dataset has verified the effectiveness of the suggested end-to-end HTR system.

Keywords: *OCR, Handwritten Text Recognition, Convolutional Layers, Bidirectional LSTMs, Attention Mechanism, Connectionist Temporal Classification, Character Segmentation, Digital Text Processing, IAM Dataset.*

2.INTRODUCTION

The usage of HMMs and RNNs, including BLSTM and MDLSTM, has enhanced handwritten word detection systems. In deep learning, there has been an ongoing debate regarding whether explicit character segmentation is necessary when utilizing CTC (Connectionist Temporal Classification) loss. However, recent studies have shown that such segmentation may be excessive when using CTC loss, ultimately providing more efficient and accurate results. Handwritten writing typically has related letters, but these methods may interpret whole sentences at once. The aforementioned methodology obviates the requirement for character-level segmentation and alignment, thereby yielding expedited and heightened precision in word identification.

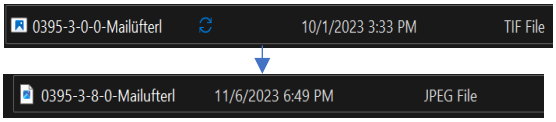
HMMs are highly effective in modeling sequences of varying lengths, although they may encounter difficulties in accurately capturing intricate details of handwriting. Recurrent Neural Networks (RNNs) have proven to be highly capable in modeling sequential data, although they do encounter challenges with vanishing gradient problems. LSTMs address these challenges and effectively manage long-range connections, although they do necessitate a significant amount of data. CNNs are highly effective in recognizing character shapes, although they may not fully capture the sequential context.

By combining them in hybrid models, with CNNs for feature extraction and RNNs or LSTMs for context modeling, researchers have found that handwriting recognition can be significantly improved. This is

achieved by using CTC loss for word alignment without the need for explicit character segmentation.

3. DATA PREPROCESSING

The first stage of the data preparation pipeline is transforming the dataset's file format to the more prevalent JPEG format. This translation method ensures uniformity and interoperability with commonly used image-processing libraries and tools. For the sake of uniformity and clarity, file names are converted to ASCII Unicode.



Subsequently, labels are generated using the picture file names, facilitating the association of images with their corresponding classes or categories. To summarize, the data is organized in a CSV file with distinct columns for labels and picture file locations. By using this technique, the dataset is customized for the specific requirements of machine learning, ensuring both ease of use and uniformity.

The dataset contains a sequence of four digits that were not correctly extracted. To address this issue, we use a regular expression pattern, namely r'(\d{4})\.jpeg$', to extract four consecutive digits that precede the ".jpeg" file extension at the end of an image file path.

Handwritten words are originally in a textual form, but machine learning algorithms want numerical data as input. The function facilitates the process of assigning a numerical value to individual characters, thereby allowing for more detailed analysis and interpretation. The encoding process effectively preserves the fundamental characteristics of each character present in a given word. The process of feature engineering is of utmost importance in facilitating the model's ability to distinguish and understand a wide variety of handwritten words and letters.

```
for char in str(text):
    idx = alphabets.find(char)
    dig_list.append(idx if idx != -1 else alphabets.find('-'))
```

4.DATA AUGMENTATION

As a data-space solution to the issue of limited data, data augmentation is implemented. Data augmentation comprises a collection of methodologies designed to augment the magnitude and caliber of training datasets to facilitate the construction of more effective Deep Learning models.

For Deep Learning models to be successful, the validation error must reach a point of convergence with the training error. Augmenting data is a very efficient approach to do this. The enhanced data will decrease the difference between the training and validation sets, as well as any future test sets, by involving a broader range of potential data points. See Fig 2.

Data augmentation is used to enhance the efficacy of machine learning models by mitigating overfitting. Overfitting arises when a model excessively grasps the intricacies of the training data, leading to an inability to effectively apply its knowledge to novel data.

Data augmentation addresses this issue by presenting the model with varied instances of data, thus enabling it to learn more general patterns rather than overfitting to the training dataset's peculiarities.

The practical application of data augmentation in this project is seen through several techniques. Basic image modifications such as horizontal flipping, color space adjustments, noise injection, shear transformation, and random cropping are employed. These techniques simulate different real-world variations that handwritten texts might exhibit, preparing the model for the complexities it will face in practical applications.

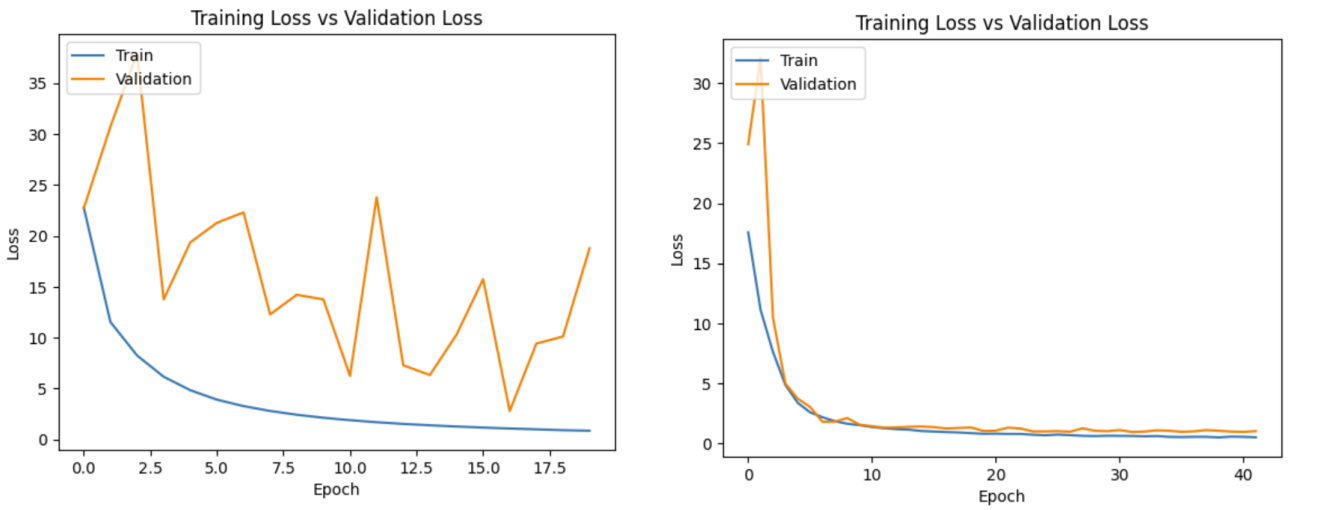
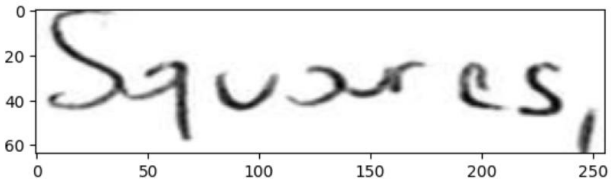
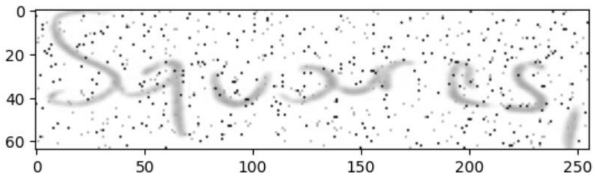


Fig 1. *Left* represents Train Loss vs Validation Loss Before applying Data Augmentation. *Right* represents Train Loss vs Validation Loss After applying Data Augmentation

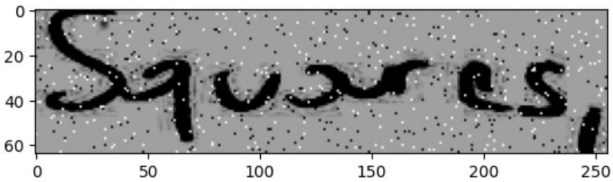
The first evidence of the efficacy of Data augmentation is derived from basic modifications like horizontal flipping, color space augmentations,



I. Adding random contrast and brightness.



II. After Salt and pepper Noise Injection



III. Finally Normalizing Pixels

Fig 2. Illustrating the sequential steps involved in the use of picture data augmentation.

noise injection, shear transformation and random cropping. These transformations encapsulate several of the invariances already addressed, which pose difficulties for picture recognition tasks.

The validation data serves as an indicator of the training procedure of the model's highest level of generalizability. Enriching the validation dataset is advisable if you expect to encounter diverse real-world data that can be replicated through data augmentation.

Many different processes may lead to the same fundamental picture deterioration known as salt and pepper noise, whereby only a small number of pixels are affected. The result is akin to seasoning a picture with a dash of salt and pepper in a pattern of white and black dots.

Quantize every pixel to B bits using the customary procedure. The pixel's value can be expressed as

$$X = \sum_{i=0}^{B-1} b_i 2^i.$$

Salt-and-pepper noise is a type of interference in images that is characterized by the presence of white and black pixel clusters. This kind of noise is known for having a heavy-tailed distribution. To illustrate this, we can consider a simple model where the original image is represented by f(x, y), and the image affected by salt and pepper noise is q(x, y).

$$\Pr[q = f] = 1 - \alpha$$

$$\Pr[q = \text{MAX}] = \alpha / 2$$

$$\Pr[q = \text{MIN}] = \alpha / 2,$$

where MAX and MIN are the highest and lowest values of the picture, respectively. The modified pixels appear as monochrome specks scattered over the picture. See Fig 3.

The expression,

(image - tf. reduce_mean(image)) / tf.math.reduce_std(image)

performs mean subtraction and standardization on image data. The process involves scaling the pixels of a picture using their standard deviation and then centering them around their mean value. This helps to enhance the clarity and quality of the image. This gives picture data a mean of zero and a standard deviation of one, which aids machine learning model training. Standardized data simplifies optimization, improves model convergence, and provides uniform feature sizes, helping models discover meaningful patterns.

5.MODEL

The initial layer of the model commonly known as the input layer, is designed to process grayscale pictures with dimensions of 64x256.

Several convolutional and pooling layers are traversed by the model as it processes the input images. These layers collaborate to extract features from the data, reduce the size of the space, and generate additional feature maps. The model learns to identify trends and distinguishing characteristics in the images by following these steps. A balance between feature maps and spatial dimensions is needed to maintain the model successful and produce accurate predictions.

After multiple "sequential" modules cascade, increasing the number of feature maps and diminishing their spatial dimensions. In image recognition tasks, hierarchical feature extraction is commonly used to gather both low-level and high-level information.

The tensor with shape (None, 64, 256) is transformed into a tensor with shape (None, 64x512) in the reshape layer. This data reshaping method streamlines the underlying data structure, making it easier to interpret feature maps quickly and accurately.

These features are refined by the model via a dense layer that is passed over the flattened feature maps. The 64-neuron dense layer generates a 64-dimensional output tensor by processing the 64x512 input tensor. The enhanced feature vector has the potential to be applied to a multitude of subsequent tasks, including object detection and image classification.

Batch normalization, dropout, and dense layer regularization are used to maintain model

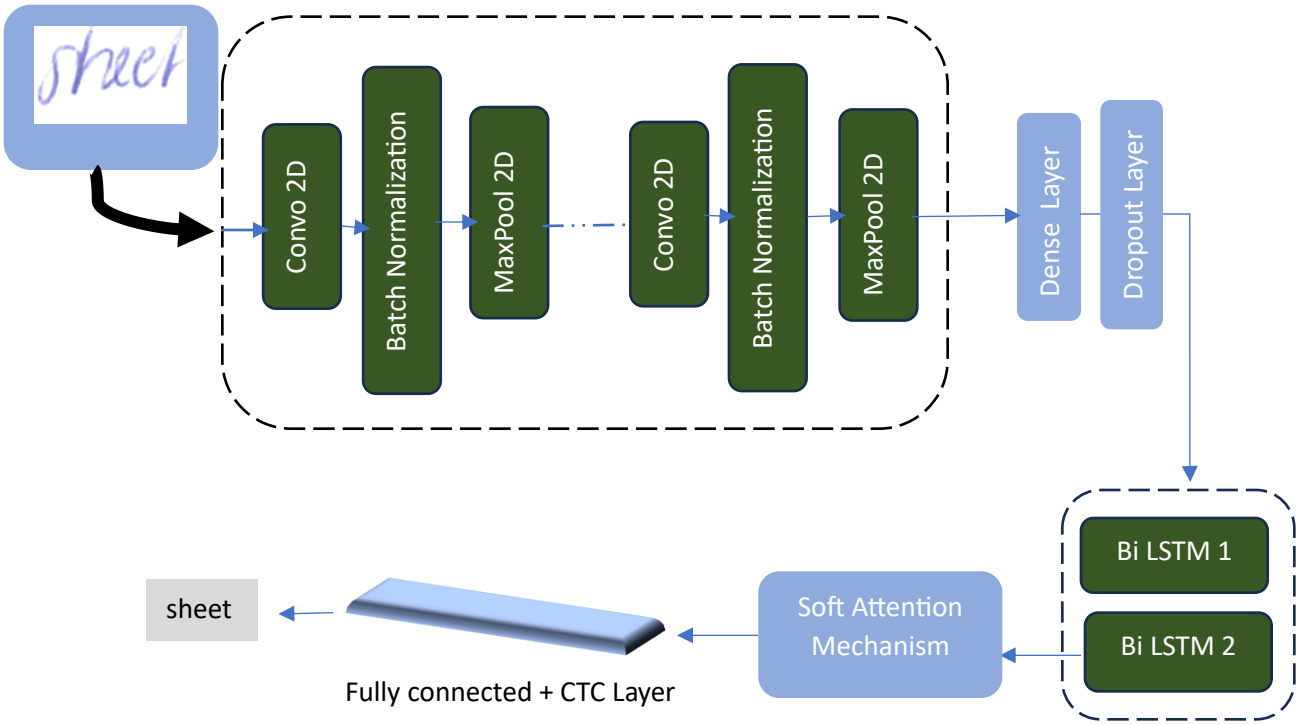


Figure 3. A neural network model comprises a Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSTM) layers, and an Attention Mechanism.

effectiveness. Batch normalization stabilizes training by normalizing layer input features, improving model convergence and robustness. During training, the dropout layer randomly sets a fraction of input units to 0. This prevents overfitting. Finally, dense layer regularization discourages complex models, making them more generalizable and robust.

A procedure denoted as an attention model necessitates a context c and n arguments y1,..., yn (where yi denotes hi in the preceding examples). The algorithm generates a vector z that is designed to function as a "summary" of the yi, placing particular emphasis on data that is linked to the context c. More formally, the weighted arithmetic mean of the yi is calculated, where the weights allocated to each yi are determined by its significance in the given context c.

Different types of attention mechanisms are essential in deep learning models. Soft attention uses learned weights to sum input elements. Hard attention chooses the highest-weighted input sequence element. Additive attention applies a feedforward network to weight calculation, multiplicative attention models interactions directly, self-attention considers sequence relationships, and transformer models use scaled dot-product attention to capture element dependencies.

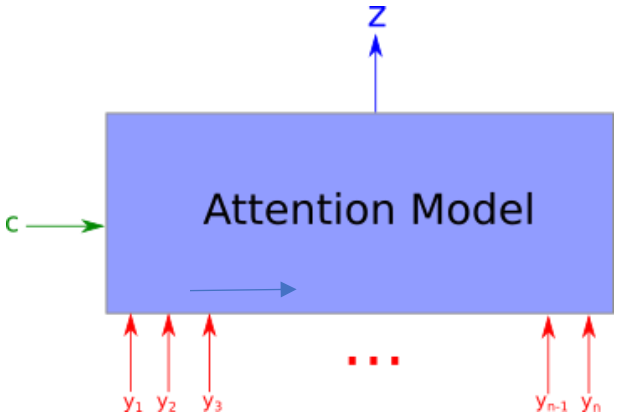


Figure 4. Represents Attention Mechanism Model

The models can selectively focus on relevant information, improving performance in sequence and variable-length data tasks.

Our model incorporates soft attention, which assigns continuous weights to input elements, enabling the model to simultaneously focus on multiple elements. Soft attention mechanisms possess differentiability, rendering them more amenable to optimization through gradient-based techniques.

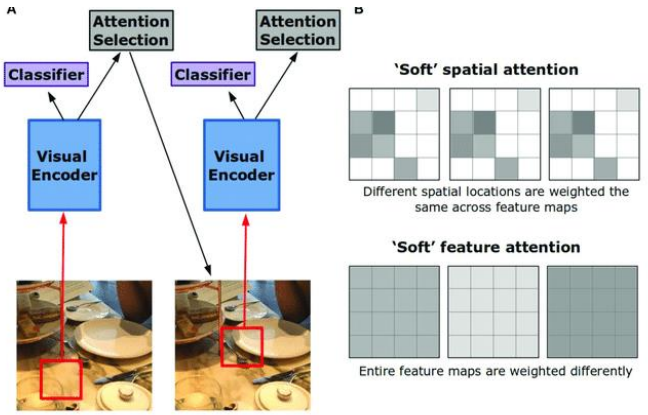


Figure 5. Represents the working of Soft Attention Mechanism

Sequence-to-sequence handwriting recognition models need bidirectional and attention layers. Contextualizing and highlighting sequence elements requires these layers. Combining bidirectional layer outputs and the attention mechanism, the concatenate layer represents the input sequence. The "dense" layer generates character class probabilities from this combined output.

The model takes non-core inputs. "label" for ground truth labels, input length for input sequence length, and label length for ground truth label length. These inputs help the model adapt and predict data accurately during training and inference.

6. RESULTS

When evaluating a handwriting recognition model, it is important to consider two key metrics: Word Error Rate (WER) and Character Error Rate (CER).

The WER, also known as the Word Error Rate, is a metric that calculates the minimum number of changes required to transform the hypothesis text into the reference text. The calculation involves dividing the combined number of substitutions, deletions, and insertions made in the hypothesis text by the total number of words in the reference text.

The WER is commonly referred to as the Levenshtein distance for words.

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

When the WER is zero, it means that the reference and hypothesis are the same.

The CER also works in a similar way, but instead of words, it considers the number of characters.

$$CER = \frac{S_C+D_C+I_C}{N_C}$$

Our model improved significantly with an attention mechanism. The Character Error Rate (CER) dropped from 5.30% to 3.98%, showing the model's improved alignment and sequence generation accuracy. The Word Error Rate (WER) also dropped from 12.09% to 9.75%. Adding attention mechanisms has improved the model's focus and

Layers	CER%	WER%
CNN+BiLTSM+CTC	5.30	12.09
CNN+BiLTSM+Attention +CTC	**3.98**	**9.75**

Table1.The significance of the attention mechanism is highlighted in an ablative study. The least error-prone parts are highlighted in bold.

context-awareness, making predictions more accurate and meaningful See Table 1.

CNN+BiLTSM+Attention +CTC	CER%	WER%
Validation Set	3.98	9.75
Test Set	4.48	10.77

Table2.The table shows the CER and WER on both the validation set and the test set.

During the evaluation phase, the performance remains strong, but there is a slight increase in the number of mistakes. The Character Error Rate (CER) goes up to 4.48%, and the Word Error Rate (WER) increases to 10.77%. This difference could be due to the test set being more challenging or containing examples that were not well-represented in the training data.

The model demonstrates robust generalization from the validation to the test set, consistently exhibiting effectiveness in accurately transcribing letters and sentences. The model architecture exhibits resilience, as seen by the relatively low rates of error.

7.CONCLUSION

This project report on "Ink to Digital: Handwritten Word Recognition" encapsulates a significant advancement in OCR technology, particularly in the recognition of handwritten text. The report detailed the evolution from traditional approaches like Hidden Markov Models and Recurrent Neural Networks to more sophisticated techniques involving Bidirectional Long Short-Term Memory networks and Connectionist Temporal Classification layers.

Key to this advancement was the shift in handling sequence-to-sequence challenges and the effective preprocessing of data, ensuring the seamless conversion of images to a uniform format and the translation of textual content into numerical data for efficient machine learning processing.

The core of the project focused on an innovative model architecture, combining convolutional layers, pooling layers, bidirectional LSTMs, and attention mechanisms. This model adeptly interpreted and digitized handwritten texts without the need for explicit character segmentation, showcasing the power of advanced neural networks in OCR.

The project not only highlighted the technical capabilities of the model but also set a foundation for future advancements in the field. It demonstrated the potential of integrating sophisticated neural networks in OCR, paving the way for more accurate, efficient, and versatile text recognition systems, thereby opening new horizons for digital text processing and analysis.

8.REFERENCES

1.Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is
wrong with scene text recognition model comparisons? dataset and model analysis.
In: IEEE International Conference on Computer Vision. pp. 4715–4723 (2019)

2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

3. Abate, S. T., Tachbelie, M. Y., & Schultz, T. (2020). Multilingual acoustic and language modeling for ethio-semitic languages. Proc. Interspeech 2020, 1047-1051. https://10.21437/Interspeech.2020-2856

4. Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. Speech Communication, 56, 85-100. https://doi.org/10.1016/j.specom.2013.07.008

5. Vaessen, N. (2020). JiWER: Similarity measures for automatic speech recognition evaluation. Retrieved from: https://pypi.org/project/jiwer/

6.Chaudhary, K., & Bali, R. (2021). Simplifying text recognition using only 1D convolutions. Proceedings of the Canadian Conference on Artificial Intelligence.

7.Ingle, R. R., Fujii, Y., Deselaers, T., Baccash, J., & Popat, A. C. (2019). A scalable handwritten text recognition system. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE.

8.Bluche, T., & Messina, R. (2017). Gated convolutional recurrent neural networks for multilingual handwriting recognition. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), IEEE.

9.Ptucha, R., Such, F. P., Pillai, S., Brockler, F., Singh, V., & Hutkowski, P. (2019). Intelligent character recognition using fully convolutional neural networks. Pattern Recognition, 88, 604-613.

10.Yousef, M., Hussain, K. F., & Mohammed, U. S. (2020). Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. Pattern Recognition, 108, 107482.

11.Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1251-1258.

12.Kang, L., Riba, P., Rusiñol, M., Fornés, A., & Villegas, M. (2020). Pay attention to what you read: Non-recurrent handwritten text-line recognition. arXiv preprint arXiv:2005.13044.

13.Hernandez Diaz, D., Qin, S., Ingle, R., Fujii, Y., & Bissacco, A. (2021). Rethinking text line recognition models. arXiv preprint arXiv:2104.07787.

14.Li, M., Lv, T., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2021). Trocr: Transformer-based optical character recognition with pre-trained models. arXiv preprint arXiv:2109.10282.

15.Michael, J., Labahn, R., Grüning, T., & Zöllner, J. (2019). Evaluating sequence-to-sequence models for handwritten text recognition. 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE.

16.Pham, V., Bluche, T., Kermorvant, C., & Louradour, J. (2014). Dropout improves recurrent neural networks for handwriting recognition. 14th International Conference on Frontiers in Handwriting Recognition, IEEE.

17.Voigtlaender, P., Doetsch, P., & Ney, H. (2016). Handwriting recognition with large multidimensional long short-term memory recurrent neural networks. 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE.

18.Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7132-7141.

19.Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., Gulati, A., Pang, R., & Wu, Y. (2020). ContextNet: Improving convolutional neural networks for automatic speech recognition with global context. arXiv preprint arXiv:2005.03191.