Paper

Preliminary Evaluation of Convolutional Neural Network Acoustic Model for Iban Language Using NVIDIA NeMo

Steve Olsen Michael, Sarah Samson Juan, and Edwin Mit

Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia

https://doi.org/10.26636/jtit.2022.156121

Abstract—For the past few years, artificial neural networks (ANNs) have been one of the most common solutions relied upon while developing automated speech recognition (ASR) acoustic models. There are several variants of ANNs, such as deep neural networks (DNNs), recurrent neural networks (RNNs), and convolutional neural networks (CNNs). A CNN model is widely used as a method for improving image processing performance. In recent years, CNNs have also been utilized in ASR techniques, and this paper investigates the preliminary result of an end-to-end CNN-based ASR using NVIDIA NeMo on the Iban corpus, an under-resourced language. Studies have shown that CNNs have also managed to produce excellent word error (WER) rates for the acoustic model on ASR for speech data. Conversely, results and studies concerned with under-resourced languages remain unsatisfactory. Hence, by using NVIDIA NeMo, a new ASR engine developed by NVIDIA, the viability and the potential of this alternative approach are evaluated in this paper. Two experiments were conducted: the number of resources used in the works of our ASR's training was manipulated, as was the internal parameter of the engine used, namely the epochs. The results of those experiments are then analyzed and compared with the results shown in existing papers.

Keywords—acoustic modeling, automated speech recognition, convolutional neural network, CNN, under-resourced language, NVIDIA NeMo.

1. Introduction

Acoustic models are one of the most important components of automated speech recognition (ASR) systems. Acoustic model functions represent the relationship between an audio signal and the phonemes or other linguistic units that make up speech in an ASR. A good acoustic model will help an ASR recognize speech inputs accurately and will produce excellent training data. Current state-of-the-art (SOTA) acoustic models rely primarily on end-to-end artificial neural model's algorithm [1] which helps simplify the conventional module-based ASR system into a single deep learning framework.

In the case of low-resourced languages, the ability to transcribe the language accurately is much more critical, since low-resourced languages have limited amounts of speech data available. As mentioned by Chuangsuwanich *et al.* [2], a speech recognizer typically requires the following: thousands of hours of transcribed speech for the statistical learning of the acoustic model, a phonetic pronunciation dictionary that determines how words of the language are decomposed into smaller phone-like units, and a large collection of texts to create the language model. For limited-resourced languages, these items are expensive and time-consuming to obtain [3]. Hence, an acoustic model is required to be able to perform excellently, although with limited resources only in the case of low-resourced languages.

The first objective of this paper is to investigate how an end-to-end CNN-based ASR model performs on the Iban corpus, an under-resourced language. Experiments have been conducted using different amounts of resources on a CNN model to identify whether this architecture has the means to overcome the problem of the lack of training data of an under-resourced language. The second objective is to perform sensitivity analysis concerning the selected architecture parameters, in order to investigate their impact on the word error rate (WER) produced. A further experiment is conducted after the most optimal amount of resources to be used for model training has been identified, with one of the parameters used in the training phase being manipulated.

2. Literature Review

2.1. Neural Network and Under-resourced Languages

Over the past few years, artificial neural networks (ANNs) have been among the most well-known options for developing an ASR's acoustic model [4]. There are several variants of ANNs in existence, including deep neural network (DNNs), recurrent neural networks (RNNs), and convolutional neural networks (CNNs). Currently, many re-

searchers are looking into the capabilities of CNN when working on ASR technologies, because this approach has shown some promising results in terms of performance, with a WER of 0.0295 using a beam search decoder with an external neural language model [5] and of 0.019/0.041 using ContextNet, a fully convolutional encoder that incorporates global context information into convolution layers by adding squeeze-and-excitation modules, with language model [6]. Several research projects have already been conducted to investigate how to further enhance the performance of an ASR using a neural network by experimenting with the acoustic model for under-resourced languages.

Implementation of a multilingual neural network ASR architecture for an under-resourced language is a popular method for achieving better accuracy. Biswas *et al*. has implemented this to improve the performance of their ASR for the native language of South Africa, isiZulu, which has an issue of "code switching", i.e. a method of combining their native language with different languages in their everyday conversations [7]. Researches implemented three types of acoustic models, with two of them implemented as multilingual methods, in order to alternatively solve this matter. They were able to enhance the performance of their ASR reaching an overall WER of ≥ 0.55 from the initial baseline of 0.63, which is an increase of roughly 0.08 in terms of reduced error rate, for each experiment they conducted. Their acoustic model was a variant of the Gaussian hidden Markov model (GMM/HMM), multilingual DNNs and multilingual time delay-long short-term memory neural networks (TDNN-LSTMs). This effort was a good improvement for their ASR in terms of WER performance, although their results are still scaled as a "high-error rate" ASR performance. They concluded that a higher amount of monolingual data for training is necessary to achieve very good WER performance for their under-resourced language.

Furthermore, He *et al.* also conducted a research experiment on improving the WER of an ASR for an under-resourced language, Iban, by adopting the multi-task learning (MLT) and acoustic landmark method for the purpose of ASR training [8]. The authors used an ASR trained with the TIMIT corpus using the MLT method and then further trained it for detection and classification landmarks. The same ASR was then cross-lingually adapted to the Iban corpus as a secondary task. From their experiment, they were able to increase the performance of an ASR trained with very few resources, "Iban 10%", meaning that ASR was trained with as little as 40 min of Iban data, to a reduced WER of 0.087 on a mono-phone and 0.0617 on a triphone ASR. Conversely, although a cross-language landmark detector provides useful information complementary to orthographic transcription, visual inspection indicates that a cross-language landmark detector is not as accurate as a same-language landmark detector. They later suggested training a more accurate landmark detector using RNN methods that could be applied to multilingual trained corpora as a means to overcome the aforementioned belief as their future work.

2.2. CNN as the Chosen Architecture

Although producing an ASR for low-resourced language has proven to be quite a challenge, past research has shown that an ASR with a single deep learning framework acoustic model, such as CNN, was able to produce promising results using only raw unprocessed speech data in estimating phonemes [9] and ASR with an excellent WER (in comparison with other ANNs), such as the DNN model architecture [10].

Palaz in [9] has proven that CNN was able to outperform or produce a similar result when compared with an ANN-based system that takes standard cepstral features as inputs. He was able to prove that contrary to his previous findings, in which poor ASR performance was attained using raw speech signal as input to DNN, his study on large vocabulary speech recognition (LVCSR) indicates that CNNs have an edge over DNNs in modeling raw speech signals. From his study, he was able to get a WER of 6.7 using a neural model of three CNN layers with one max pooling layer plus a hidden layer and HMM as decoder, but taking only a raw speech signal as its input, in comparison with HMM-ANN with one hidden layer and using mel-frequency cepstrum (MFCC) to preprocess its input with a WER of 7.0. Both were tested and trained on the Wall Street Journal corpus. As the number of layers increases to three hidden layers for each model, Palaza's CNN model achieving a WER of 5.6 outperforms the ANN model even further, as the latter achieved a value of 6.4. It is known that under-resourced languages lack the resources needed for training. Furthermore, even without the need of data pre-processing, having only raw speech data as a requirement for an ANN to train its model, would serve as a potential step in perhaps not eliminating, but reducing the difficulties faced while producing well-performing ASR for under-resourced languages. This may be achieved by allowing only raw speech data as a necessity to train an ASR, eliminating the need for an abundant resource for preprocessing and accurate transcribing.

Moreover, much research has been conducted recently showing that CNN outperforms several other architectures, including the famous speech recognition neural network architecture, RNN, in building ASR for low-resourced languages. In papers [11]–[14], Reyes *et al.*, Thai *et al.*, Mon, and Lekshmi and Sherly stated that CNN offers a truly exceptional feature extraction capability. These authors were conducting experiments on their respective under-resourced languages using the CNN architecture. Since CNN is a SOTA architecture for image and graphic classification, it is not surprising that this NN would be able to perform better in terms of feature extraction. Taking raw speech signal as a spectrogram, CNN was able to distinguish even the smallest differences in features, leading to an accurate prediction of spoken words. Mon in [13] also mentioned that optimization of the hyperparameter in CNN architecture was capable of producing better performance than other ANNs, as well as greatly impacted the perfor-

mance of ASR through the setting of the number of feature maps and pooling the sizes of the CNN. Conversely, Thai *et al*. stated that both deep RNN and GMM/HMM models were outperformed by their fully convolutional acoustic model which yields significant accuracy improvements [12]. The same authors further mentioned that CNN mitigates the issue of incompatibility in parallelization on modern hardware, as cited by Collobert *et al*., and is advantageous as it requires only a few parameters to collect sufficient important features for an accurate prediction that leads to the reduction of computational cost [15]. Furthermore, Lekshmi *et al*. implored that CNN is a better deep learning model because of its capability to reduce spectral variations [14] and to model spectral correlations between signals, as reported in Rabiner and Juang [16]. All the results and benefits mentioned above further support the choice of CNN.

Finally, another reason why CNN was chosen is due to the Iban language being relatively unexplored as a CNN-ASR model. There are not many works on CNN for Iban. One of the most recognizable ones was that by Juan who experimented on the Iban language using the DNN architecture and GMM/HMM [17]. Hence, this also serves as a motivation for CNN to be studied to address this research gap. Focusing on the manipulation of data amount and the parameters that exist in the CNN architecture, this study aims to look into the architecture's performance on the basis of the two factors that were highlighted earlier when dealing with the same under-resourced situation in the case of the Iban language.

The rest of this paper is organized as follows. Section 3 discusses the architecture of CNN. Section 4 presents the methodology used for the experiment. Section 5 describes the protocol used in the setup, as well as the results that were obtained. Sections 6 and 7 discuss the findings and constraints of this experiment, while Section 8 concludes the research.

3. CNN Model Architecture

Rather than using fully linked hidden layers, as its comparable variant, i.e. DNN does, CNN uses a unique network structure that comprises alternating convolution and pooling layers. The idea behind the CNN architecture was adopted from the pattern of connections between neurons in the human brain and from the arrangement of the visual cortex [18]. In the case of a CNN, the main purpose of convolution is to extract features from visual inputs. Such an approach is highly useful in image processing and identification. Through the learning of im-age features using small squares of input data, convolution is capable of preserving the spatial relationship between pixels, as an image is basically a matrix of pixel values [19]. Essentially, a matrix of pixel values can be used to represent every image as well as waveforms. Typically, CNN comprises four major operations: (1) convolution, (2) non-linearity – rectified linear unit (ReLU), (3) pooling step, and (4) classification (fully connected layer).

For the first operation (convolution), let us begin by considering a 5-image matrix with pixel values assuming, in a special case, values of 0 and 1 only, as shown in Fig. 1.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Fig. 1. Original 5×5 matrix.

We shall also consider another 3 matrix as shown in Fig. 2. This small matrix is known as a "kernel", "filter" or "feature detector".

1	0	1
0	1	0
1	0	1

Fig. 2. 3×3 Kernel matrix.

As shown in Figs. 3–5, imagine that our 3×3 matrix (blue selected cells) slides over the 5×5 image matrix, moving by one pixel at a time. This process is known as a "stride". As the slide takes place, element-wise multiplication is computed between the two matrices and its multiplication outputs are added to obtain the final integer, which forms a single element of the output matrix (green cells).

The output matrix (green highlights) was formed by sliding the 3×3 kernel matrix over the original 5×5 matrix. The computation of the dot product is a process called "convolved feature" or "activation map" or the "feature map". Note that the kernels detect features from the original input image (5×5 matrix). On the same input matrix (image), different feature maps are produced by various values of the kernel matrix. During implementation, the values of these kernels are learned by the CNN throughout the training process. However, parameters such as the number of

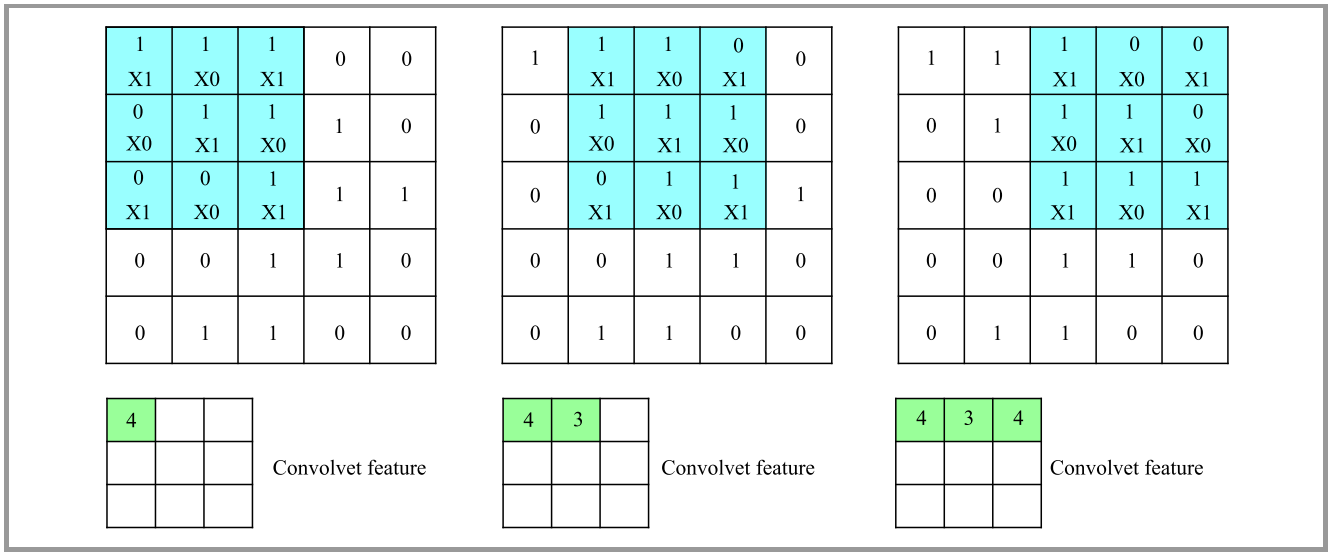

Fig. 3. First to third strides of the kernel matrix.

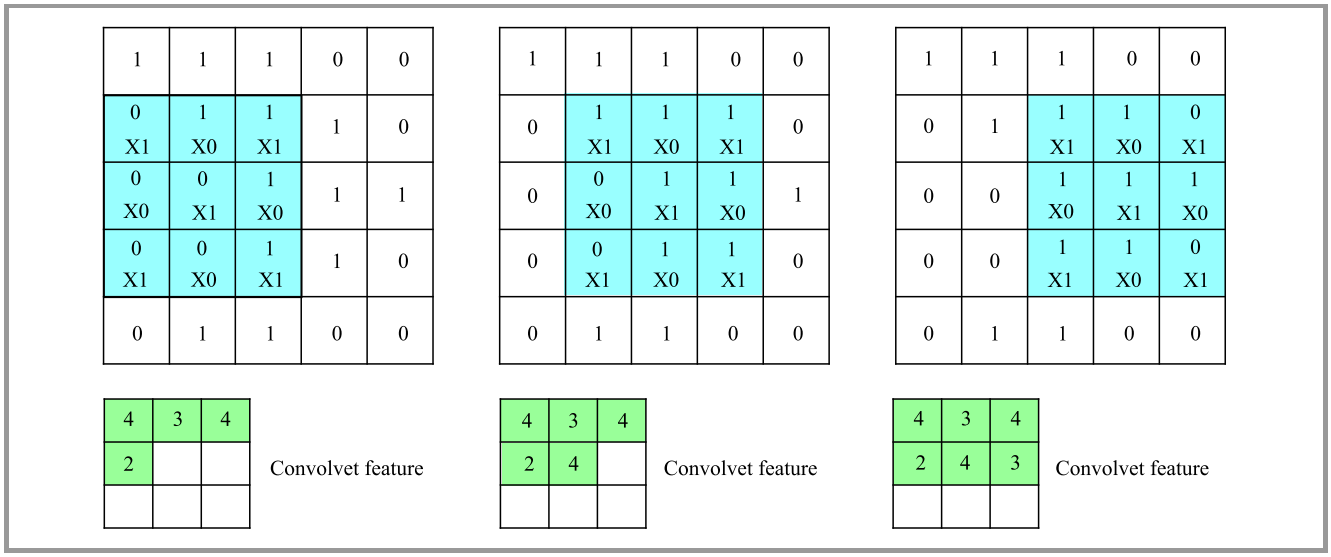

Fig. 4. Fourth to sixth strides of the kernel matrix.

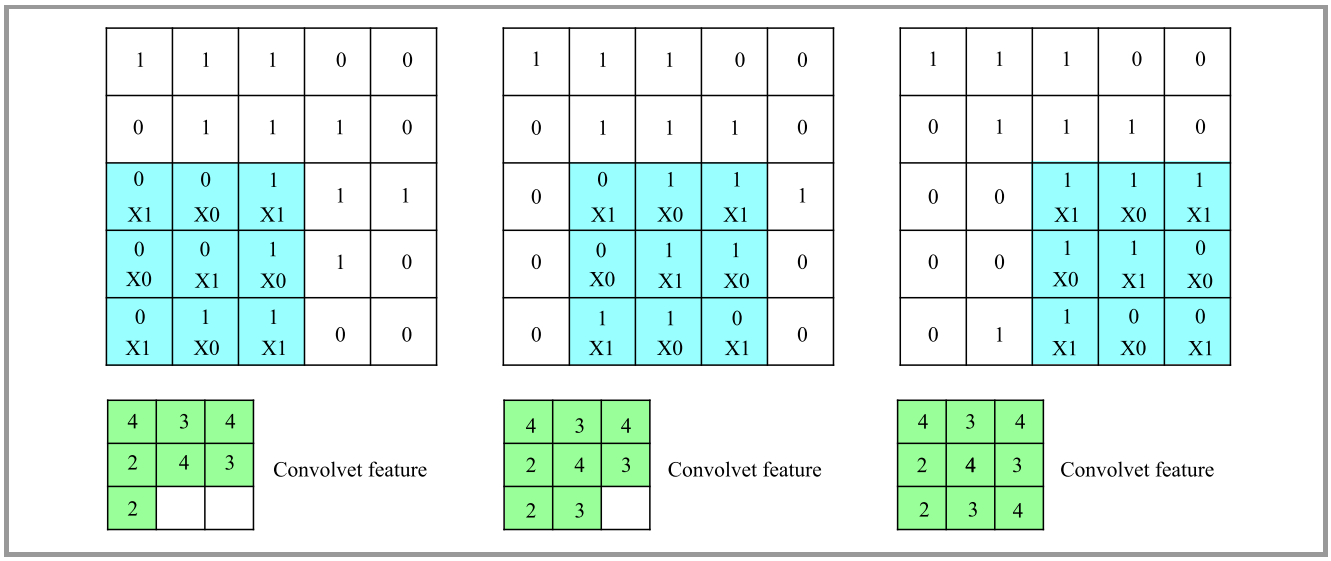

Fig. 5. Seventh to ninth strides of the kernel matrix.

kernels, kernel size, and architecture of the network still need to be specified before the training process. Moreover, before the convolution step is performed, three parameters have to be specified, as they control the size of the convolved feature. These include: depth, which represents the number of kernels used in convolution; stride, representing the number of pixels that were slid over the input matrix by our kernel matrix; and zero-padding, which involves zeros that pad around the border of the input matrix, allowing the application of the kernels to the bordering elements of the input matrix. A higher number of kernels corresponds to extraction of a higher number of image features and to the network's better recognition of patters in unseen images.

Next, while proceeding with the second operation (non-linearity – ReLU of CNN), non-linearity should be applied to CNN, since convolution is a linear operation but most real-world data are non-linear. The purpose of the activation function is to transform the summed weighted input from the node into the activation of the node or output for that input of a neural network [20]. For CNN, ReLU is the most commonly used activation function, as it has been found to perform better in most situations when compared with the other non-linear functions such as tanh or sigmoid. Also, ReLU has become a common activation function for several types of neural networks, as it offers ease of the training process and often ensures better performance. It is a piece-wise linear function that outputs the input directly if it is positive, and the 0, if it is negative [20]. From the previous example, when an image was formed from the output feature map, it would be a mixture of black and white shades since it is a result of black = negative values and white = positive values. By applying ReLU to CNN, we would be removing all negative values from the feature map and would only be taking the non-negative values. This allows the CNN to identify the image of a feature map, as there are only non-negative values to indicate the differences of layers in the feature map. A feature map to which ReLU has been applied is also called a "rectified" feature map.

The third operation, namely the pooling step (spatial pooling), also known as sub-sampling, is to reduce each feature map's dimensionality while simultaneously retaining important key information. Usually, the pooling layer comes after finishing the convolutional layer. It serves as a layer for reducing the input volume's spatial dimensions (width $\times$ height) for the next convolutional layer, although the depth dimension of the volume is not affected. This operation is also known as subsampling because the loss of information occurs as well, alongside the reduction of dimensions. This loss however is more beneficial to CNN, mainly because of the following:

- the next layers in the network will suffer from lower computational overheads,
- overfitting will be prevented.

Similarly to convolution, the pooling layer takes a sliding window in which serves as a another newly defined matrix window that is to be striding along with the rectified feature map. In this operation, the window will be transforming the values from the feature map by taking either the maximum value or the average value, as observed under the window known as "max pooling". Because of its better performance characteristics, max pooling has been preferred in most situations. Figure 6 depicts the operation of max pooling on a rectified feature map using a 2×2 window.

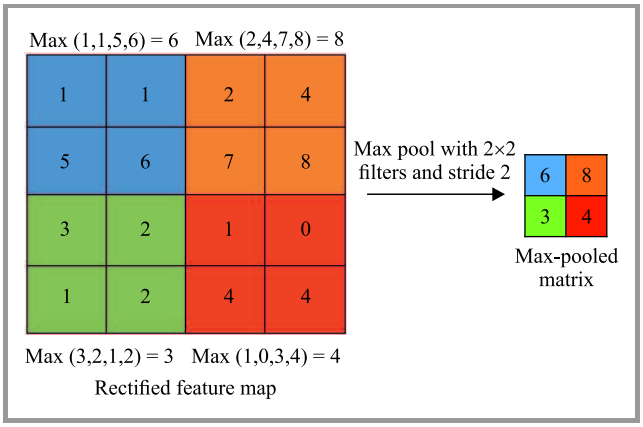

Fig. 6. Max pooling of a rectified feature map.

Max pooling can be deduced as an operation of applying a function over the input values. No new parameters were introduced during this phase and it uses fixed-sized portions at a time, with a size configurable as a parameter. Note that although CNN may undergo max pooling, pooling operations are excluded in most CNN architectures. Thus far, the operations have reduced the size of the original matrix making it even smaller than it was before, since it was divided into smaller-sized feature maps. As the first operation progresses further to the second convolution, the second pooling step, and then onward to the third convolution and also the third pooling step and so on, the size of the matrix will be reduced further and more feature maps will be produced, until the network can finally conclude one or more possible outputs for the original image.

Finally, as mentioned earlier in this section, the fourth operation (classification or a fully connected layer) is just as what its name implies – the CNN layers are all fully interconnected with each other. In other words, each neuron in the preceding layer is connected to each neuron in the next. Outputs from the second and third operations represent high-level features of the original image. The fully connected layer uses these features for classifying the input image into various classes on the basis of the training dataset [19]. Furthermore, implementation of fully connected layers enhances the learning of non-linear combinations of those features. A combination of the features gained through the convolution and pooling step would be very helpful in classifying the image. Through the use of the Softmax function, the sum of the output probabilities is ensured to be equal to 1. Softmax, as an activation function in the output layer of a fully connected layered

network, takes a vector of arbitrary real-valued scores and compresses it to a vector of values between 0 and 1 that sums to 1.

4. Methodology

4.1. Architecture of CNN Used

The ASR experiment involving an under-resourced language and the CNN architecture is performed using the NVIDIA Neural Module (NeMo) in Google Colab. NeMo is a flexible Python toolkit that allows building SOTA speech and language deep learning models by using reusable building blocks that can be safely connected together for conversational AI applications [21]. NeMo allows the configuration of the architecture to be laid out by specifying the model using a Yet Another Markup Language (YAML) file. This allows us to have flexibility in designing our CNN architecture. The content of the YAML config file has an entry labeled as "encoder" with a field called "jasper", with parameters that specify the criteria of one block, as such:

filters = 128, repeat = 1
kernel = [11]
stride = 2
dilation = 1
dropout = 0.2
residual = false
separable = true
se = true
se_context_size = -1.

Our configuration follows the Jasper_4x1 model, which comprises (K = 4) blocks of single (R = 1) sub-blocks and a greedy connectionist temporal classification (CTC) decoder. A total of seven blocks was used, with the first six of them consisting of pointwise and one-dimensional time channel separable convolutional layers, followed by batch normalization, and then the ReLU layer. Two subsequent invariants of K $\times$ R blocks have 256 and 1024 filters each, respectively, and both were also repeated once. Moreover, NeMo works closely with libraries from Pytorch Lightning (PTL), in which the training and saving of models and checkpoints had relied heavily upon PTL functions to complete.

The framework is shown in Fig. 7. Typically, the experiment requires us to include raw audio data files together with a manifest file that comprises metadata of our audio files as inputs. NeMo requires a standardized manifest file in which it is shown that each line in the file corresponds to one audio sample, such that the number of lines in the manifest is equal to that of samples that are represented by that manifest. Moreover, NeMo specifies that a line must contain the path to the audio file, the corresponding transcript or path to the transcript file, and the duration of the audio sample.

The CNN model then produces an output of text files that hold the details and the WER of the trained dataset that was calculated using a third-party plugin, the speech recognition scoring toolkit (SCTK).

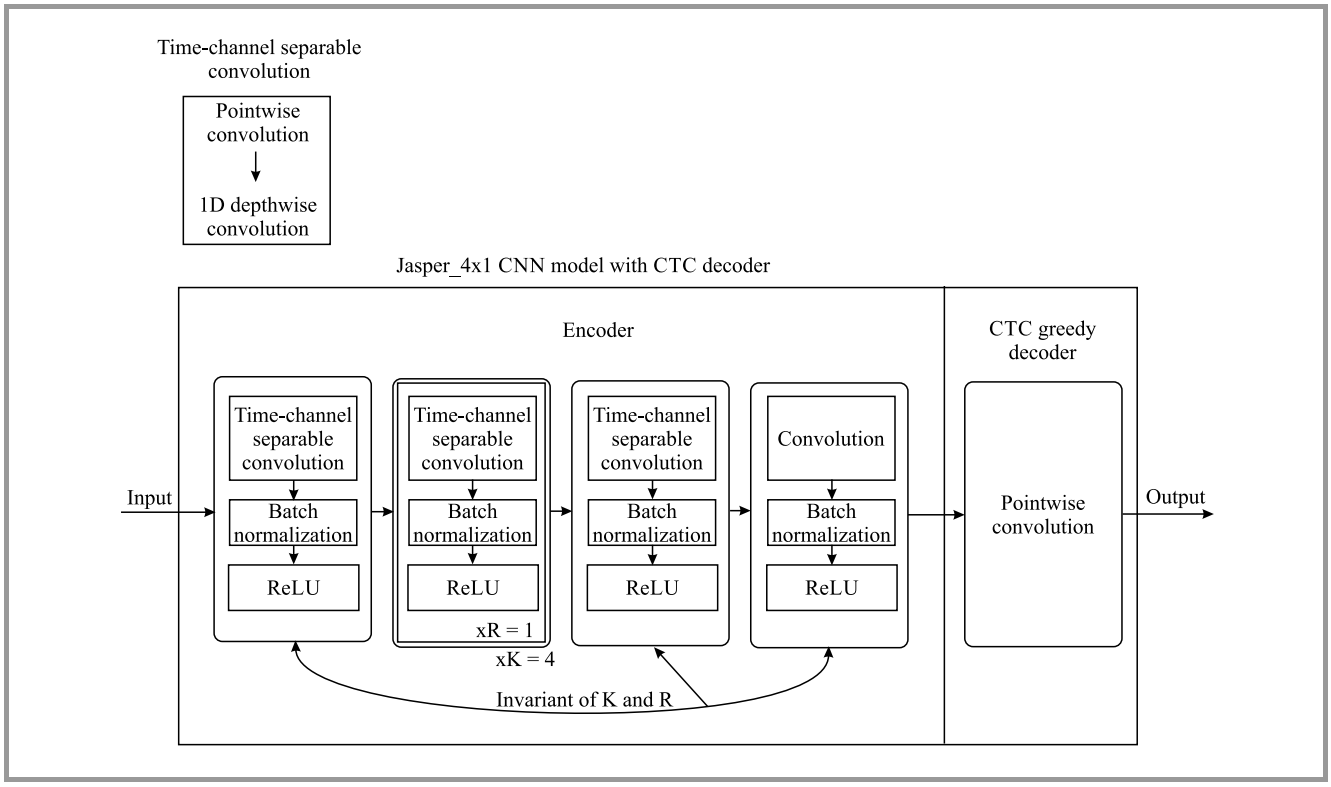

Fig. 7. Framework model of the CNN architecture used.

4.2. Iban as an Under-resourced Language

The under-resourced language that was used is the Iban language, a native language from Sarawak, Malaysia which is still widely used by local people for communication and official procedures. A total of 400 lines of Iban sentences have been used for testing, whereas another 2000 lines of sentences taken from the original corpus dataset are used for training. The resources were previously gathered by Juan *et al.* who used them to investigate the resourcefulness of closely related languages for automatic speech recognition in low-resource languages from Malaysia [22].

The corpus was composed of news gathered after receiving the permission to collect data from a local news station in Malaysia, Radio Televisyen Malaysia Berhad. A total of 3 GB of audio data was received for pre-processing, but only 1.1 GB could be relied upon, as the rest turned out to be of poor quality. The news data were transcribed into text during a data collection workshop organized at the Faculty of Computer Science and Technology, Universiti Malaysia Sarawak, which was aimed to have users with basic skills in using a transcriber application to annotate speech signals [17]. Eight native speakers were hired to produce the transcription data.

The Iban corpus has a total speaker of 6, a male-to-female ratio of 1:2, 473 sentences, 11,000 words, and a total of 71 min of audio data for its test set. For its train set, there were 17 speakers in total, with a male-to-female ratio of 7:10, 2659 sentences, 61,000 of words, and a total of 408 min audio data. In total, 3000 sentences uttered by 23 speakers in 8 h of clean speech were gathered during the workshop [17].

5. Experiment Setup and Results

To investigate the authors' first objective, i.e. how an end-to-end CNN-based ASR model performs on the Iban language, using a different number of lines of sentences for testing and training, this experiment evaluates the number of resources needed for a CNN architecture to perform well on an under-resourced language such as Iban, since mostly, resources are lacking. Experiment 1 comprises two rounds. The first round of the experiment was conducted to identify the most optimal number of training lines to be used in order to train our model to get the best WER performance on an under-resourced language. With just 10 testing lines, Table 1 shows the summary of the first round of the experiment.

Table 1
Protocol for the training lines experiment

No. of training lines	Total speakers	Total female speakers	Total male speakers	Total lines spoken by female	Total lines spoken by male
200	10	5	5	100	100
400	10	5	5	200	200
600	10	5	5	300	300
800	11	6	5	400	400
1000	13	7	6	500	500
1200	13	7	6	600	600
1400	15	8	7	700	700
1600	16	9	7	800	800
1800	16	9	7	900	900
2000	17	10	7	1000	1000

5.1. Experiment 1

The 10 testing lines used for this experiment were spoken by a total of five speakers, of whom three were female and two were male. Five lines were spoken by both male and female speakers. The number of lines was distributed randomly among the speakers, but the total lines spoken by the two genders were kept equal. Table 2 shows the protocol of the testing line used in the training line experiment. Table 3 shows the WER result achieved by a range of 200-2000 training lines with 10 test lines only.

Table 2
Protocol for the testing line used in the training lines experiment

No. of testing lines	No. of female speakers	No. of male male speakers	No. of lines spoken by female	No. of lines spoken by male
10	3	2	5	5

Table 5 shows the WER result achieved by a range of 100–400 testing lines with 2000 training lines.

We found that the most optimal number of training lines to be used to get the best WER performance would be 2000 lines for training with a WER of 0.832. Subsequently, the second round of Experiment 1 was conducted to obtain baseline results for an increasing number of testing lines. As obtained from the previous round, using 2000 training lines as the most optimal number of training lines to get the best WER, Table 4 shows the protocol of the second round of the experiment. Table 5 shows the resulting WER. Results showed that the best WER performance is 0.868 when tested on 400 lines.

5.2. Experiment 2

To fulfill the second objective of this paper, namely to verify sensitivity of the architecture to selected parameters in order to investigate their effects on WER produced, we further analyze the selected parameter that affects the performance of the CNN-based ASR model. We used the most optimal setup obtained from the previous experiment, with 2000 and 400 training and testing lines. We fine-tune the number of epochs for each setup. This is done to ana-

Table 3
WER result for the training lines experiment with 10 testing lines only

Training lines	200	400	600	800	1000	1200	1400	1600	1800	2000
Test lines	10	10	10	10	10	10	10	10	10	10
Epochs	100	100	100	100	100	100	100	100	100	100
WER	1.00	1.00	0.94	0.95	0.94	0.94	0.91	0.90	0.89	0.83

Table 4
Protocol for testing lines experiment

No. of testing lines	No. of female speakers	No. of male male speakers	No. of lines spoken by female	No. of lines spoken by male
100	4	2	50	50
200	4	2	100	100
300	4	2	150	150
400	4	2	200	200

Table 5
WER result for the testing lines experiment with 2000 training lines

Test lines	100	200	300	400
Training lines	2000	2000	2000	2000
WER	0.88	0.87	0.91	0.87

lyze whether CNN models are capable of further improving performance of the ASR model on under-resourced languages by adjusting its parameters. The setup focuses on manipulating the number of epochs from 500 to 5500, and the models will be tested by measuring WER.

The numbers for both testing and training line protocols were maintained the same as in Experiment 1 – 400 testing lines and 2000 training lines.

Table 6 shows the result of WER achieved by the model under the epochs of 500–5500, using 400 and 1000 testing and training lines, respectively. The table shows that the lowest WER achieved is 0.67, which was attained using the ASR model when the epoch was set at 5500.

As for the time taken for the training process to finish, we discover that each increase of the number of epochs by 500 is equivalent to 3–4 h of training. Let us assume that 500 epochs take 3 h to complete the training process. Then, 1000 epochs, i.e. double the original amount, will take 6 h to complete. Therefore, it is estimated that the longest time taken for training was 44 h with the setup for 5500 epochs.

6. Discussion

6.1. Analysis of Results

Table 3 shows that as the number of the training lines increases, the performance of ASR evaluated based on its WER increases as well. This is also true for the increase in the number of testing lines used, as shown in Table 5. This indicates that even an end-to-end CNN architecture is limited by the requirement of the resourcefulness of a given language. The resourcefulness of the language still plays a vital role in training a CNN ASR to perform well on an under-resourced language. The number of lines for testing and training in both setups clearly shows that it is not enough for the ASR to perform excellently, as WER remains above 0.8 in both setups. Nonetheless, the results show that they could be further improved with more training data. This proves that the resourcefulness of the setup with a higher number of lines of sentences used for training the ASR using CNN on an under-resourced language helps achieve better performance in terms of accurate audio transcription. Under-resourced languages have small amounts of data for training, and it seems that even CNN could not overcome this problem. This answers the first objective of this experiment.

Furthermore, the data presented in Table 6 shows that an increase in the number of epochs also increases performance, as WER improves. An epoch is the number of passes of the entire training dataset that the deep learning algorithm has completed. The experiment batches comprise iterations. As such, an epoch is not the number of iterations completed, but the number of batches that have been completed from the entire training session. Table 6 shows that there is a pattern related to the performance of the training process. The early stage of Experiment 2, starting from an epoch setup of 500, shows the minimum performance of an outstanding WER of 0.85, which then started to decrease gradually at a rate of -0.2 to -0.4 WER, until it reached the epoch setup of 2500. At the epoch setup of 3000, performance of the experiment starts to increase very slightly, at a rate of -0.01 WER only, leading us to conclude that the effort to manipulate the number

Table 6
WER results for Experiment 2

Epoch	500	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500
WER	0.85	0.81	0.78	0.75	0.73	0.71	0.70	0.69	0.68	0.68	0.67

epochs for a performance upgrade may have reached its limit.

The peak performance of the ASR has been achieved when the epoch count was set at 5500 with a WER of 0.67. At this stage, the results showed that we were going in the direction of overtraining the model, although this claim must be investigated further. This may also be affected by the number of resources used to train the model, as it limits the increase in performance of the proposed ASR model. Hence, for a CNN architecture, a high number of epochs may help, for the betterment of WER, it may not be the most effective variable to optimize the model's performance. the CNN architecture still relies on the amount of resources to be trained in order to produce a very good ASR for under-resourced languages.

6.2. Comparison of Results with a Similar Study on Iban ASR

Juan has also researched the Iban language using DNN, one of the most commonly applied ANN algorithms when generating ASRs [17]. Using DNN, the author was able to get a WER performance of 0.184 on a 7 h speech audio data, which is a much better results for an ASR model than the one achieved with a CNN model. This may be caused by integrating a pronunciation dictionary developed during in the course of the research project, and by a language model that was trained on 2 the 2 million-word Iban news data using the SRI language modelling toolkit (SRILM) to their ASR model. Our CNN model was an end-to-end variant, where neither the usage of a pronunciation dictionary nor language model is a necessity, in contrast to the traditional method of ASR development.

The absence of a pronunciation dictionary may have affected the performance of the developed model, as the difference of 0.486 WER exists when one compares the CNN end-to-end model without a pronunciation dictionary with Juan's DNN model with both a pronunciation dictionary and a language model [17]. Her pronunciation dictionary was developed using the Malay grapheme-to-phoneme (G2P) model, which helped them to produce an Iban G2P. The method subsequently enabled them to produce a hybrid G2P as well – a model that incorporates pronunciation rules from Malay and Iban. The G2P was used to produce a pronunciation dictionary for the architecture of their Iban ASR.

The existing pronunciation dictionary may help the model recognize specifically the phonetic structure of Iban language, thus increasing its WER performance, whereas our CNN model depends wholly on the performance of the CNN-based acoustic model ASR and on the learning of phonetics through convolutions of frames.

7. Limitations of Experiments

An obvious hindrance that we have been struggling with while conducting this research experiment was the vagueness and ambiguity in understanding how the NVIDIA NeMo ASR engine works. Although it is intended to serve as a toolkit for building new SOTA conversational AI models, we believe that this new engine also has a high potential of becoming a "go-to" engine as an ASR development toolkit. Unfortunately, to date, its documentation is limited, and we are unable to fully benefit from this tool. For example, the technique for integrating the language model and pronunciation dictionary, the process of saving and loading checkpoints, and the method for obtaining output transcripts and WER files are still lacking and incomplete. This may be due to the fact that this engine remains rather new, as it was only released in 2019. Thus, it is still undergoing development, fortunately at a rapid rate. Nonetheless, this has placed upon us a blockade that limits us from fully configuring our setup the way we have planned it initially. However, we still encourage the usage of this engine to help its developer gain more feedback for improvement purposes.

8. Conclusion and Future Work

This paper has shown how an end-to-end CNN architecture using the NVIDIA NeMo engine and the Google Colab toolkit performs on raw under-resourced language data (Iban), with a WER of 0.67, using only 2000 lines of sentences for training and without the help of a pronunciation dictionary and a language model. The CNN architecture may not be able to eliminate the lack of resources that is faced by under-resourced languages, but with the integration of other tools, such as language models, this architecture may also be able to achieve an excellent WER. This paper presents also the results and a discussion on gradually increasing the performance of the model with different configurations on its number of epochs.

Future work should include the training of the model using a more considerable amount of data, since it will be interesting to see what the minimum amount of data is, needed for the model to train in order to be able to achieve a very good WER performance, under the same setup and weights. Moreover, an increase in the number of filters or blocks in the CNN architecture may yield different, possibly better results. Also, it would be desirable to identify whether the performance of the model upon exceeding 5500 epochs would help the ASR increase its performance, would keep it stagnant or maybe would even lead to the production of a negative output, namely to an increase in WER caused by overtraining. Furthermore, other parameters such as the learning rate, which would help increase the WER of the model when configured to a certain state, must be identified. Finally, the integration of language models and/or pronunciation dictionaries with the model should be investigated to learn whether it could help increase the transcription accuracy of the CNN ASR model on the Iban language.

Acknowledgement

This work was supported by the Malaysia Comprehensive University Network (Grant number: GL/F08/MCUN/11/2020). We are grateful for the support enabling us to complete the project.

References

[1] V. Passricha and R. Aggarwal, "Convolutional neural networks for raw speech recognition", *IntechOpen.*, vol. 32, pp. 137–144, 2013 (DOI:10.5772/intechopen.80026).

[2] E. Chuangsuwanich, "Multilingual techniques for low resource automatic speech recognition", Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2016 [Online]. Available: http://hdl.handle.net/1721.1/105571

[3] B. Pulugundla *et al.*, "BUT system for low resource Indian language ASR", in *Proc. 19th Ann. Conf. of the Int. Speech Commun. Assoc. Interspeech 2018*, Hyderabad, India, 2018, pp. 3182–3186 (ISSN: 1990-9772).

[4] O. Mamyrbayev *et al.*, "Voice identification using classification algorithms", in *Intelligent System and Computing*, Yang Yi, Ed. IntechOpen, 2020 (DOI: 10.5772/intechopen.88239).

[5] J. Li *et al.*, "Jasper: An end-to-end convolutional neural acoustic model", in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc., Interspeech 2019*, Graz, Austria, 2019, pp. 71–75, 2019 (DOI:10.21437/Interspeech.2019-1819).

[6] W. Han *et al.*, "ContextNet: Improving convolutional neural networks for automatic speech recognition with global context", in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc., Interspeech 2020*, vol. 2020-Octob, pp. 3610–3614, 2020 (DOI: 10.21437/interspeech.2020-2059) [Online]. Available: https://arxiv.org/pdf/2005.03191.pdf

[7] A. Biswas, F. D. Wet, E. V. D. Weisthuizen, E. Yilmaz, and T. Niesler, "Multilingual neural network acoustic modelling for ASR of under-resourced English-Isizulu code-switched speech", in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc., Interspeech 2018*, Hyderabad, India, 2018, pp. 2603–2607 (DOI: 10.21437/Interspeech.2018-1711).

[8] D. He, B. P. Lim, X. Yang, M. Hagesawa-Johnson, and D. Chen, "Improved ASR for under-resourced languages through multi-task learning with acoustic landmarks", *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc., Interspeech 2018*, Hyderabad, India, 2018, pp. 2618–2622 (DOI:10.21437/Interspeech.2018-1124).

[9] D. Palaz, R. Collobert, and M. Magimai-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks", in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc., Interspeech 2013*, Lyon, France, 2013, pp. 1766–1770 [Online]. Available: https://arxiv.org/pdf/1304.1018

[10] D. Palaz, M. Magimai-Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal", in *Proc. IEEE Int. Con. on Acoust., Speech and Sig. Process. ICASSP 2015*, South Brisbane, QLD, Australia, 2015, pp. 4295–4299 (DOI: 10.1109/ICASSP.2015.7178781).

[11] F. Reyes, A. Fajardo, and A. Hernandez, "Convolutional neural network for automatic speech recognition of Filipino language", *Int. J. of Adv. Trends in Comp. Sci. and Engin.*, vol. 9, no. 1.1, pp. 34–40, 2020 (DOI:10.30534/ijatcse/2020/0791.12020).

[12] B. Thai, R. Jimerson, R. Ptucha, and E. Prud'hommeaux, "Fully convolutional ASR for less-resourced endangered languages", in *Proc. of the 1st Joint Worksh. on Spok. Language Technol. for Under-res. Lang. (SLTU) and Collab. and Comput. for Under-Resourced Lang. (CCURL)*, Marseille, France, 2020, pp. 126–130 [Online]. Available: https://aclanthology.org/2020.sltu-1.17.pdf

[13] A. N. Mon, "Myanmar language continuous speech recognition using convolutional neural network (CNN)", Ph.D. thesis, University of Computer Studies, Yangon, 2019, pp. 87–88 [Online]. Available: https://meral.edu.mm/record/4316/files/AyeNyeinMonThesisBook.pdf

[14] K. R. Lekshmi and E. Sherly, "An acoustic model and linguistic analysis for Malayalam disyllabic words: a low resource language", *Int. J. of Speech Technol.*, vol. 24, pp. 483–495, 2021 (DOI: 10.1007/s10772-021-09807-1).

[15] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2Letter: an End-to-End ConvNet-based Speech Recognition System", arXiv:1609.03193v2, 2016.

[16] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ: Prentice-Hall, 1993 (ISSN: 9780130151575).

[17] S. S. Juan, "Exploiting resources from closely-related languages for automatic speech recognition in low-resource languages from Malaysia", PhD. thesis, Université Grenoble Alpes, France, 2015, pp. 115–118 [Online]. Available: https://tel.archives-ouvertes.fr/tel-01314120/document

[18] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way", Towards Data Science, 2018 [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[19] Ujjwal Karn, "An intuitive explanation of convolutional neural networks", the data science blog, 2016 [Online]. Available: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets

[20] J. Brownlee, "A gentle introduction to the rectified linear unit (ReLU)", Machine Learning Mastery, 2010 [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/, 2020.

[21] "NVIDIA Deep Learning NeMo Documentation", Nvidia website, 2021 [Online]. Available: https://docs.nvidia.com/deeplearning/nemo/index.html

[22] S. S. Juan, L. Besacier, B. Lecouteux, and M. Dyab, "Using resources from a closely-related language to develop ASR for a very under-resourced language: A case study for Iban", in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc., Interspeech 2015*, Dresden, Germany, 2015 (DOI: 10.21437/Interspeech.2015-318).

Steve Olsen Michael received his B.Sc. in Computer Science and Information Technology (Software Engineering) from Universiti Malaysia Sarawak, Malaysia in 2019. Since 2021, he has been a M.Sc. student focusing on speech technology, automated speech recognition (ASR) systems, convolutional neural network (CNN) models, and acoustic modeling, working at the Faculty of Computer Science and Information Technology (FCSIT), UNIMAS. His current research interests are in acoustic modeling techniques using CNN, ASR for under-resourced languages, and speech recognition toolkits.

https://orcid.org/0000-0002-2510-5736
E-mail: steveolsen.ms@gmail.com
Universiti Malaysia Sarawak
94300 Kota Samarahan
Sarawak, Malaysia

Sarah Samson Juan presently works at the Faculty of Computer Science and Information Technology at Universiti Malaysia Sarawak. She received her Ph.D. from Grenoble-Alpes University, France, after completing her thesis on automatic speech recognition systems for under-resourced language in Malaysia. Her research interests focus primarily on natural language processing techniques and machine learning approaches relied upon to develop acoustic models, language models and pronunciation dictionaries.

https://orcid.org/0000-0002-9590-1666
E-mail: sjsflora@unimas.my
Universiti Malaysia Sarawak
94300 Kota Samarahan
Sarawak, Malaysia

Edwin Mit is an Associate Professor at the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak. He received his Ph.D. in Software Engineering from the University of Salford, UK. His research interests is focus on modeling quality software and the formal approach.

https://orcid.org/0000-0002-1670-8586
E-mail: edwin@unimas.my
Universiti Malaysia Sarawak
94300 Kota Samarahan
Sarawak, Malaysia