Automatic Multilingual Code-Switching Speech Recognition

Nguyen Tuan Anh, Dang Thi Hien, Nguyen Thi Hang

Abstract - **In this study, an efficient yet accurate end-to-end multilingual Code-Switching Speech Recognition model has developed, allowing direct conversion of raw speech audio signals into text of multiple languages. This single system for multiple language aims to eliminate the use of each model for a language, in order to increase the ability to share features between languages, minimize the latency of hybrid systems and it can be extended to other objects. Unlike the single-language Automatic Speech Recognition (ASR) model that uses coding of characters or words, the multilingual model applies the same encoding to all languages. However, the vocabulary is encoded into a numerical dictionary and partitioned for each language. The single end-to-end system is designed to directly convert multi-lingual raw audio to dictionary of Unicode numbers of words of languages, which is mapped 1:1 into text of the corresponding language. This method allows to expand to an unlimited number of languages, furthermore, it identifies languages automatically without the need for a separate model. This model uses word pieces, as opposed to graphemes, to reduce the modeling unit gap in multiple languages. The proposed network has been validated on Chinese and Vietnamese, demonstrating a significant improvement of accuracy in comparison with other single and multi-lingual models' techniques in monosyllabic and multi-tone languages.**

I. INTRODUCTIONS

Automatic Multilingual Code-Switching Speech Recognition is a method of converting speech into text, using multiple languages, and each sentence can contain multiple languages spoken together. With end-to-end (E2E) technique and word coding, the whole speech utterance is fed into the model. The accuracy of the model is assessed according to the True labels and predicted labels, so it is not necessary to have experts in separate language fields. That means the model doesn't need to prepare data including things like language-specific acoustic, language model, phoneme, pronunciation lexicon, also known as grammar knowledge free, while performance is constantly improving compared to single-language E2E systems [1]–[4].

Many previous models have many limitations when working with Multilingual ASR ([5]–[8]). Some phrase recognition on multilingual keyword spotting ([9], [10]) for multilingual is also studied. Studies on Acoustic Models (AMs), some models have focused on the study of common phone sets ([5], [6]), some others have designed models with share parameters ([11],[12]). A design noteworthy in models is that some of the lower layers of the Deep Neural Network (DNN) are shared between languages and the output layer is language-specific ([8], [11], [13]). Traditional models often require language-specific Pronunciation Models (PMs) and Language Models (LMs). Therefore, while inference must know speech language identity ([8]). In addition, AMs, PMs and LMs are usually optimized independently so errors can occur during training that are difficult to control [2].

Language expandable is the ability to expand the number of languages shared in a model. Currently many mono-language models are designed for each specific language and cannot be expanded, or it is difficult to train a new language. With a technique designed without the need for a language expert, in order to add a new language to the system, the only thing needed to do is to prepare the corresponding speech audio and human readable label.

When the system works in multiple languages, to ensure accurate recognition, determining which speech belongs to which language is important in the final text output. From there, it is also possible to identify the similarity in the word pronounced, the syllables next to each other in the sentence. For traditional methods, to determine the type of language needs a separate model [1]. Based on segmented paging technique, our model can automatically classify languages, which is very useful in transcribe speech for bilingual dialogue.

Code-switching can be defined as a sentence that may contain more than one language appearing together. This makes it difficult for monolingual ASR models because the Out of Vocabulary (OOV) phenomenon is caused by the vocabulary of the second language. But this way of speaking is a very common phenomenon, especially for Asian countries like Vietnam and China, mixing their mother tongue with English or the language that two people are communicating with. As in the example in Table 1.

Table 1. Code-switching language

Hello cả nhà (Hello everybody)
Lúc nào tôi 下班 thì chúng ta gặp nhau nhé (Let's meet when I'm off work)
Sorry, 我不知道 (Sorry, I don't know)

With the method of coding languages by paging, word splitting, and sharing languages on the same model is proposed in this study, the input speech containing code-switching automatically is solved.

Some main contributions of this study as follows:

- Improved the single model for monosyllabic and multi-tone in multi-languages.
- Auto language identification in the model that does not need to use multiple models for the task of language identification and content recognition.
- Language expandable, with the new label coding method, the model can add other languages to the model.

- Expert grammar knowledge free, with end-to-end model training, it is not necessary to understand grammar rules.
- Code-Switching ASR, which allows the model to identify content spoken by multiple languages in a sentence.
- Context-independent Speech Recognition.
- Imbalanced multilingual dataset processing method.
- The model works well with multi-dialect, a form of multilingual but a variation of a language [14].

II. Block-Based Residual Neural Network (BRN)

A. Language Coding and Identity

For coding, each language will be using word-based. Theoretically, the number of languages that can be coded is arbitrary. To improve the ability to compare results, this study was conducted in two languages, Chinese and Vietnamese. In Chinese, because the language does not use the alphabet, each character is considered a word, a meaningful word can be composed of several words put together, and when compounding words, there can be variations in pronunciation. From there we have two ways to encode, one is coding by each character and the second is coding by each meaningful phrase. To minimize the number of vocabulary words, this study chose word encoding, with Chinese being a single character. Here, all characters in the database are encoded:

$$page_i \rightarrow page_i + n_{w_i} \quad (1)$$

Where i refers to a language, n_{w_i} refers to the total vocabulary of that language. Similar to Vietnamese, the feature of this language is the use of alphabet system and extended characters in Unicode encoding to character encoding, word structure consists of several letters and accented markings for tones.

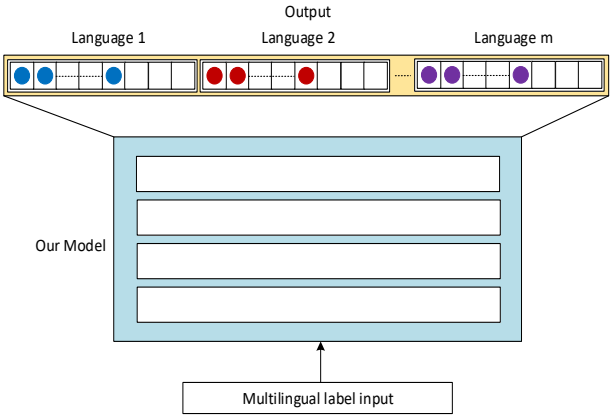

Figure 1. Conventional encoding of output labels

Each word may not have a complete meaning yet. To make sense of what people want to say, some words still need to be put together. Thus, each character in Chinese will be corresponding to the word in Vietnamese. For simplicity of coding, this study chose word coding. From there the Vietnamese data will be coded as formula (1), but located in another page (using new i for each language). Word-based is selected because it was proven that it can balance both OOV and performance issues ([15][16]). The segment pagination for languages is shown as Figure 1.

B. Imbalanced Multilingual Data Processing

This section is described research methods for balancing data in a multilingual model. Data imbalance is a common phenomenon of speakers in languages around the world. Languages with more speakers will tend to have more data. In ASR systems, an E2E multilingual model will be trained on all components, therefore, the data imbalance is very sensitive. In this section, two avenues for data balancing are explored: (1) sampling data and (2) extending the model architecture.

1) Data sampling – Up-sampling method

The Up-sampling method can be used to treat data imbalance. Low-resource languages will be up sample to a more balanced ratio. In case the data has to be balanced, all languages are up sample up to equal the most resource-rich language. For generalization, $s(i)$ is calculated using the following formula:

$$s(i) = \frac{n_i + \alpha * (n^* - n_i)}{\sum_i [n_i + \alpha * (n^* - n_i)]} \quad (2)$$

Where $n^* = \max_{i=0,\dots,n-1} L_i$, and α denoted to adjustable parameter, with $\alpha = 0$: no upsample, $\alpha = 1$: upsample to all languages for equal proportions.

In the research [17], [18], they also gave a similar method. Instead of using the AM method for traditional models, the E2E method was used.

C. Baseline Models

BRN is designed based on the ideas of the cutting-edge models, to understand the architecture of BRN, the theory of these baseline models will be shown below.

ResNet - Residual Network. [19]. In Deep Neural Network, when increasing the number of layers, it will be harder to train. According to K.He in the paper "Deep Residual Learning for Image Recognition" [19], he found that adding more layers to the network would increase training error and harder to train to achieve high accuracy. Deep Residual Convolution significantly increases the number of layers in a network. Input data is processed by layers or a block then add or concatenate with previous layer via shortcut to produce output results. Gradient in ResNet can flow directly from input to output of convolutional layer and/or blocks. On the other hand, due to the base network is convolution, so calculation speed is faster than other structure. He has shown empirical evidence to show that the ResNet network is more easily optimized and can achieve significant increases in depth. In the article “Deep Residual Learning for Image Recognition” [19], they showed that their depth reached 152 layers and won the ILSVRC & COCO 2015 Competitions, has the best results compared to only stack layer network structures. The structure of ResNet network as Figure 2

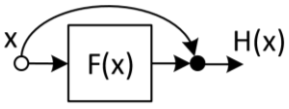

Figure 2. Residual network block

The output of the layers or a block of the network is $H(x)$, If x was fitted to label then simply set the weight to 0, otherwise, fitting $F(x)$, so we have:

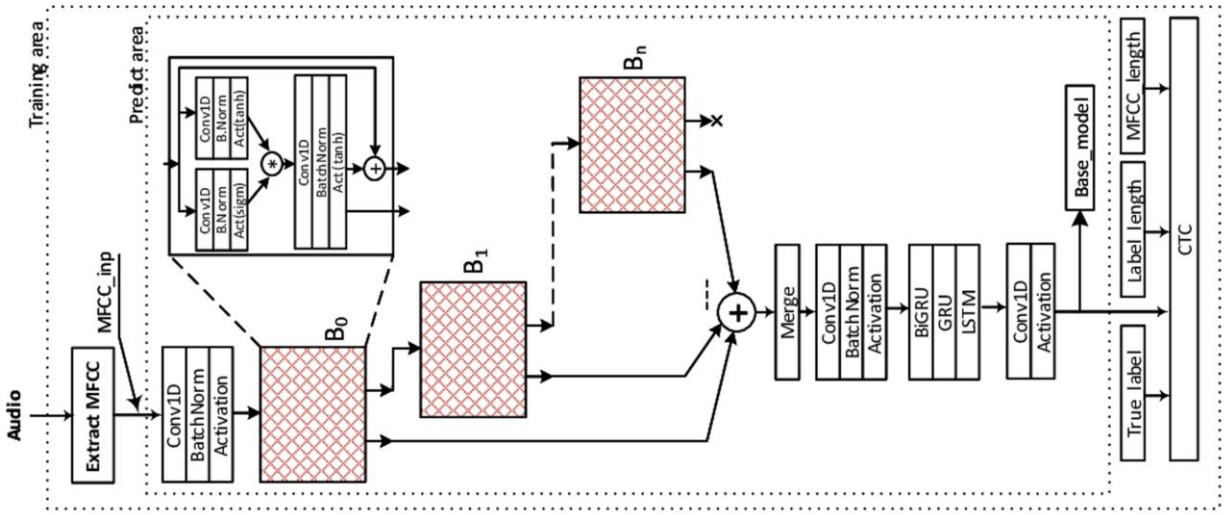

Figure 4. System Network Architecture

$$F(x) = H(x) - x \quad (3)$$

The advantage of this network is to keep features, to avoid vanilla gradient, however, due to regular gradient can only flow along the transmission network and can be flown across or combined with the features in each block by summation, which can degrade information before the end network.

DenseNets - Densely Connected Convolutional Networks [20]. The full name of this architecture is Dense Convolutional Network. With the idea that to keep more feature information through layers, DenseNets has all the layers structure that inherits information from all previous layers. It can be represented as in Figure 3 and formula (4)

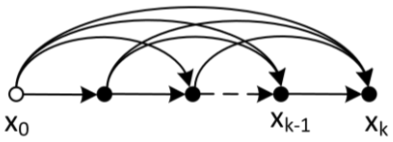

Figure 3. Dense network block

The advantage of this network is that it can explore new features from previous layers [21]. However, since all the following classes are inherited from the previous class, it leads to redundancy of information, making the model increasingly bulky and heavy.

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (4)$$

ResNeXt - Aggregated Residual Transformations [22]**.** This network is based on the principle of network modularization, by repeating building blocks and aggregating the results after having completed the transformations with the same topology.

$$B(x) = H(Wx + b) \quad (5)$$

$$F(x) = \sum_{j=1}^{C} B_i(x)$$

Since the image and speech characteristics are slightly different. In this study, an improved network from baselines is proposed to better match with sequence data such as Speech Recognition, KWS for multilingual model in Audio.

D. Block-Based Residual Neural Network (BRN)

In the ASR pre-processing, we denote X as MFCC input feature, $X = [x_0, x_1, \dots, x_{n-1}]$ where n refer to number of samples in dataset. Each feature comes with a true label, so we denote L as true labels vector, $L = [l_0, l_1, \dots, l_{n-1}]$. A label vector, $l_i, i = 0, .., n-1$, can be vectorized into $l_i = [w_0, w_1, \dots, w_{m-1}]$ where $w_j, j = 0, .., m-1$ refer to each word in the vocabulary, $m = |l_i|$, vectorized method depends on the language to choose character or word based. In this study, only keywords is focused with a given list ω, $\omega = [\omega_1, .., \omega_q]$, with q refer to number of given keywords, all words in labels vector will be treated as garbage (ω_0) if it is not in ω, so we denote $\hat{\omega}$ as KWS labels vector, $\hat{\omega} = \omega_0 \cup \omega$ and $L \in \hat{\omega}$. Noted that ω_0 cannot be treated as Null, None or "blank" (in CTC procedure) because the position of w_j maybe need to observe.

The architecture of the network is defined as shown in the Figure 4. Pre-processing step, denote as "Extract MFCC" block, creating MFCC future of utterances, saving to hard disk if it is not existed, which will help the training process faster in later step; the "Base_model" block refer to predict model. $B_k, k = 0, .., N-1$ is denoted as ResBlock[1] module which plays the role of the core module of the BRN network, N refer to number of $ResBlock * block$. Each of core block B_k is constructed by 1D Convolution, Batch Normalization and Activation layers that is calculated by the formula (6), for simplicity, b is omitted in the equation.

$$H_\delta(x) = \delta(W^T x) \quad (6)$$

where x and H refer to input and output vector of the layers considered. δ or maybe σ represents the activation type of Sigmoid or Tanh. To optimize the parameters, through practical experiments, a dropout layer is added after the Batch Normalization layer. The dropout parameters have been changed when experiments to find

[1] ResBlock: ResNet block with modified structure

the best results. The element-wise multiplication in formula (7 is used in the core block B_k.

$$F(x) = \sigma\left(W\left(H_\sigma(x) * H_\delta(x)\right)\right) \quad (7)$$

Here F is the output vector of residual mapping to be learned with input vector x. To perform shortcut connection for B_k, calculation in (8) is used by element-wise addition.

$$B_k = F(x) + x \quad (8)$$

Here the output, B_k, will be treated as input vector x of B_{k+1}, and $F(x)$ in (9) is aggregated in residual transformation as described as in ResNeXT.

$$R = H_\sigma\left(\sum F(x)\right) \quad (9)$$

where R is extracted feature output of BRN that are shared from multiple layers from input to output of the network. This is an advantage of BRN for the purpose of sharing features in multilingual KWS. Characteristics between different languages will be similar, so sharing features between multiple languages will reduce the number of network parameters. From here, to improve predictive accuracy, several RNN classes will be used.

$$\Psi = g\left(RNNs(R)\right) \quad (10)$$

Where R is the aggregated of blocks in (9). Ψ denote as "Base_model" block as shown in Figure 4, using to predict. $RNNs$ are some Recurrent Neural Network layers, including Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), which can be customized to the appropriate number of layers. The final output, Ψ, is the Probability Feature table, which will be used to calculate the loss with the CTC[23] function to update all

$$\begin{aligned} \alpha_i &= \{\,1 \mid |l_i' \cap \omega| > 2\,\} \\ \beta_i &= \{\,1 \mid |l_i \cap \omega| > 2\,\} \\ Acc &= \frac{\sum_i \alpha_i}{\sum_j \beta_j} \end{aligned} \quad (14)$$

parameters.

1) CTC based automatically speech keyword spotting.

Aiming to design the network for speech Keyword Spotting with non-aligned data sets, the final training layer is done by CTC algorithm [23], for acoustic feature sequence conversion to determine input signal have or not the keywords, if it has then point out which one. Given the probability vector Ψ of length P and the true label L, Ψ is needed to convert to predicted label $\hat{Y}$:

$$\hat{Y} = g(\Psi) \quad (11)$$

Here vector probability $\Psi = [x_0, x_1, .., x_{P-1}]$ and output predicted labels vector $\hat{Y} = [y_0, y_1, .., y_{T-1}]$ has length T, with $T \leq P$. g refer to learn-able CTC function. When training, CTC will find all possible paths to decode Ψ, each path is called $y_{\pi_t}^t$ with $\pi_t \in L'$ where $L' = L \cup blank$. Here $blank$ denotes just a keyword separation, it is not part of keyword list and it will be removed when the the best answer is chosen by CTC function. To find the paths, a CTC probability table is calculated:

$$p(\pi|X) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall t \in L' \quad (12)$$

where $y_{\pi_t}^t$ refer to a possible path, illustrated as Figure 5 in detail, see [23].

Since many paths are found, the label will look like "-aa-b-", "-a-bbbb-" or "-aaa-bbbb-" depending on the path that may be the candidate labels. From there, the repeating characters and $blank$ (-) are removed which called mapping many-to-one decoder $B: L'^T \rightarrow L^T$, where $L^{\leq T}$ is the set of possible label in $\hat{Y}$, for examples $B(-a-bbbb-) = B(-aa-b-) = ab$. It is predicted labels which will be searched by Greedy search CTC decoding.

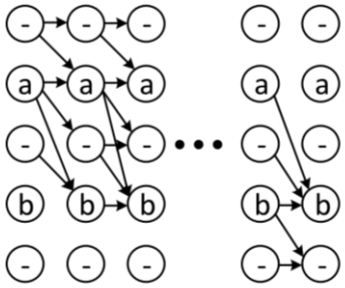

Figure 5. Illustration of the CTC forward backward algorithm applied to the labelling "ab"

2) Greedy search decoding

During experiments, both the Beam Search and Greedy Search Decoding were used. Because the Greedy Search Decoding was significantly faster, so Greedy Search Decoding is used in all models [23]. This method will produce transcription without other information.

$$\pi^* = argmax_{\pi \in N^t} \prod_{t=1}^{T} p(\pi_t|X) \quad (13)$$

Here π^* denotes as best path found. N^t refer to total possible paths in CTC probability table calculated in Figure 5. To get accuracy results, the presence of the keywords in the given keyword list $\omega = [\omega_1, \omega_2, .., \omega_n]$ is calculated:

Where α_i, β_i denote as the existing keywords in i^{th} predicted label (l') and true label (l), respectively. This method of calculating accuracy gives the same result as the false alarm method when comparing between models, but the results are more obvious, so it will be used to compare results between models.

III. Experimental Results

A. Feature Extraction

In the field of speech recognition, the feature extraction of utterances is very important. This study focused on solving low-resource problems, so small data sets of multiple languages were chosen, THCH-30 for Mandarin and ViVos for Vietnamese. Mel frequency cepstral coefficients (MFCC) are most widely used for speech because it preserves the maximum frequency characteristics of human speech. Therefore, in this study, MFCC was selected for training. To generate the MFCC coefficients, a DCT with log spectral estimation is calculated by smoothing the FFT with about 20 nonlinear frequency distributions on the spectrum, called Mel-scale frequency [24]. In this article, due to the requirements of application, the project was first trained on public dataset. Features are extracted over the entire datasets. If

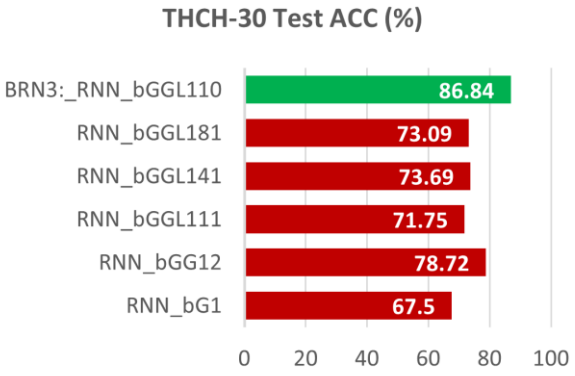

Figure 6. THCH-30 Test ACC (%), higher is better

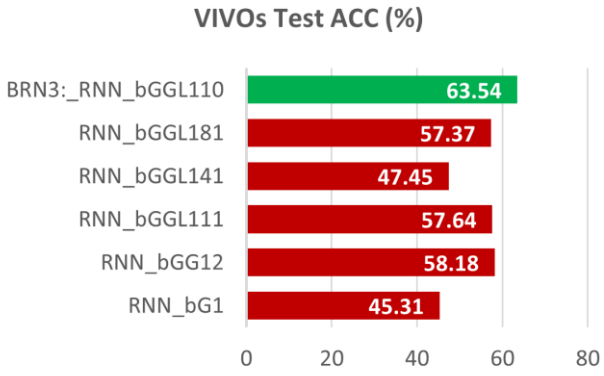

Figure 7. VIVOs Test ACC (%), higher is better

Table 1. Variations of Models

Model Name	Model Architect	Capacity	Total Params
RNN_bG1	BiGRU	64	33K
RNN_bGG12	BiGRU-GRU	64-128	132K
RNN_bGGL111	BiGRU-GRU-LSTM	64-64-64	103K
RNN_bGGL141	BiGRU-GRU-LSTM	64-256-64	411K
RNN_bGGL181	BiGRU-GRU-LSTM	64-512-64	1165K
BRN3:_RNN_bGGL110	BRN-BiGRU-GRU-LSTM	3-64-64-32	292K

required to use the model for ASR or KWS in streaming purposes, the utterance will be extracted to MFCC feature in frame of $t(ms)$ then put to model for prediction.

B. Model Architecture Configurations

The hyper-parameters are optimized by Adam optimizer, the maximum training is about 400 epochs. Batch size varies according to the hardware trained on. To achieve the highest training speed, learning rate is selected from 0.001 then reduced after 5 epochs without improved results. Each utterance input is extracted MFCC with 20-channels. The length of features is padding equal to the longest one. In this case, the maximum padding is 375. To compare the effectiveness of the models, our model and the baseline were experimented with various custom changes.

C. Results and Discussion

To be able to compare with other models, the study [25] was selected as a baseline model because of their experimental results show that LSTM models has been outperformed the feed-forward DNN and performed better compared to cross-entropy loss trained LSTM. Moreover, LSTM, BiLSTM or Multi-layer RNN architecture has been used in many KWS tasks such as ([26]–[32]).

1) Compare block-based residual network with baseline models

The results of this study are shown in Figure 6 and Figure 7. Specifically, "BRN3: _RNN_bGGL110" is the version of our model that delivers the best results compared to all base model customizations. With RNN, a carefully calibrated customization produces the best results, but it only achieves an ACC score of 80.27% for THCH-30, while for VIVO is 62.2%, still lower than BRN. To better understand the convention of each network option, the options are denoted as follows:

The capacity is the number of hidden cells (with RNN) or block (with BRN). Symbols of models are defined as follows:

- RNN: Recurrent Neural Network
- bG: Bidirectional GRU (Gated recurrent units)
- G: GRU (Gated Recurrent Units)
- L: LSTM (Long Short-Term Memory)
- BRN: Block-based Residual Convolutional Neural Network

From Figure 6 and Figure 7, we can observe that the results of BRN are higher than that of DeepRNN with the same RNN architecture and with the similar training time. For Chinese, because the dataset is about 30h, BRN shows superior results compared to baselines. The highest RNN result was 80.27% and the BRN result was 86.84%, higher than baseline 33.3%.

With Vietnamese, this dataset has nearly 15 hours, too few for a language, so all models are hard to give the best results in practice. However, to compare the BRN with the baseline, once again from Figure 7, the BRN produces a higher result, 63.54% compared to 62.2%. The author hopes that if the amount of data is increased, the models will increase accuracy. It is also a work to be done in the future.

2) Results of the variations of Block-Based Residual Network

To see the best results, the models have been carefully customized with many different configurations to ensure the best configuration is found. In Table 1, six

variants of the BRN were tested, each with different capacities, different footprints, and therefore different results.

With the parameters shown in Table 1, the results are shown in the Figure 8. It is mainly about changing the number of Residual blocks, with each configuration, trained three different models to explore the capacity of the model in that working range. The result shows that the number of Residual blocks is 3, the network BRN3: _RNN_bGGL110 still gives the highest results. With the number of Residual blocks being 4, the BRN4: _RNN_bGGL121 network gives higher results. However, accuracy will change as the amount of data changes.

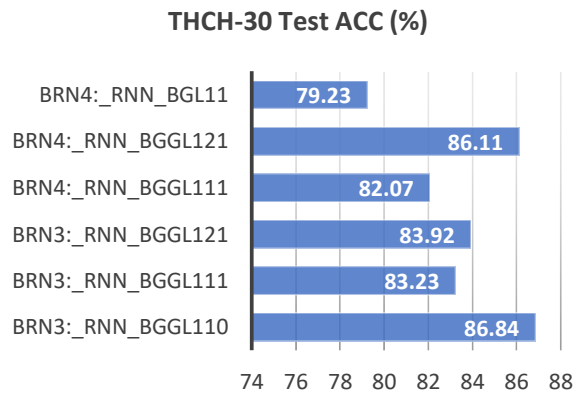

Figure 8. THCH-30 Test ACC (%), higher is better

To further illustrate that, the ViVos dataset is used, with the result of this data set, we can observe that a network with four Residual blocks has a higher result. The reason may be because in Vietnamese there are more tones than that in Chinese, on the other hand the data set is less than THCH-30, so the larger model will learn more details.

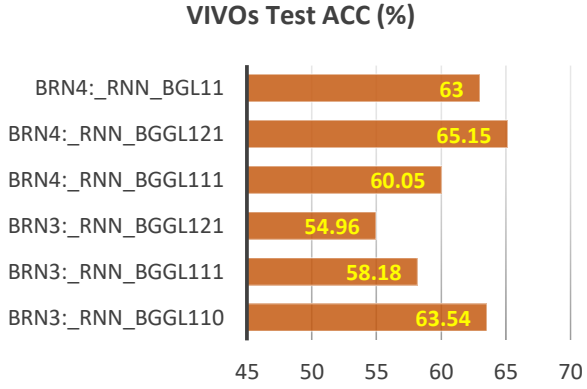

Figure 9. VIVOs Test ACC (%), higher is better

For a comprehensive comparison of models, the observation from Figure 10 shows that most BRN variants give higher results than the network variant with the RNN baseline. Because this study focuses on model research for low-resource languages. So, the architecture of the model plays a very important role to improve the accuracy of each language.

3) Discussion.

As shown in Figure 10, when using Deep RNNs with varying network depths and using BRN with the varying configurable blocks, it is easy to observe that BRN has better results in most cases. From there, the model can easily use code-switching for multiple languages or code-switching for multi-dialect languages.

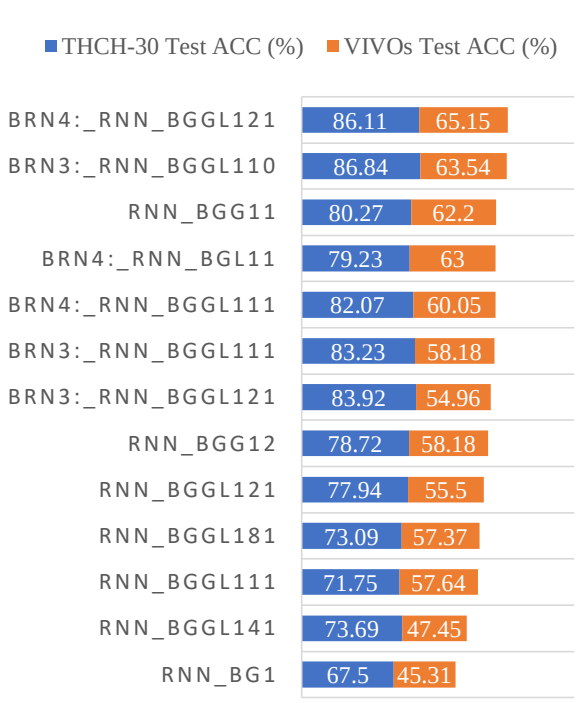

Figure 10. Compare all models, higher is better

IV. Conclusions

In this study, we revisited the ResNet, DenseNets, ResNeXT architect. Based on these cutting-edge structures, we have introduced the BRN network with design by inheriting those advantage design. The purpose of this network is improving the accuracy when detecting speech keywords in multiple datasets for multiple languages without aligned labels, in this case, the Mandarin and Vietnamese. This Keyword Spotting approach is suitable for real applications. Experiment on THCH-30 and ViVos corpus, both datasets are very modest in quantity. BRN shows that it is outperforms Deep RNNs base models. As future work, we want to expand this work to support simultaneous KWS for multi-language online recognition.

V. Acknowledgments

In this article, we would like to especially thank to Thai Nguyen University of Technology, Thai Nguyen, Viet nam for supporting us in the experimental process.

VI. References

[1] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 265–271.

[2] S. Toshniwal *et al.*, "Multilingual speech recognition with a single end-to-end model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4904–4908.

[3] M. Karafiát *et al.*, "Analysis of Multilingual Sequence-to-Sequence speech recognition systems," *arXiv Prepr. arXiv1811.03451*, 2018.

[4] J. Cho *et al.*, "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 521–527.

[5] T. Schultz and A. Waibel, "Fast Bootstrapping of LVCSR Systems with Multilingual Phoneme Sets," in *Proceedings of the 5th European Conference on Speech Communication and Technology, Vol. 1*, 1997, pp. 371–373.

[6] T. Schultz and A. Waibel, "Language Independent and Language Adaptive Acoustic Modeling for Speech Recognition," *Speech Commun.*, vol. 35, no. 1–2, pp. 31–51, 2001.

[7] T. Niesler, "Language-dependent state clustering for multilingual acoustic modelling," *Speech Commun.*, vol. 49, no. 6, pp. 453–463, 2007.

[8] G. Heigold *et al.*, "Multilingual acoustic models using distributed deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8619–8623.

[9] N. T. Anh, H. T. T. Hang, and G. Chen, "One approach in the time domain in detecting copy-move of speech recordings with the similar magnitude," *Int. J. Eng. Appl. Sci.*, vol. 6, no. 4, 2019, doi: 10.31873/ijeas/6.4.2019.05.

[10] N. T. Anh, H. T. K. Dung, and N. T. P. Nhung, "Adaptive Cross-Correlation Compression Method in Lossless Audio Streaming Compression," *Int. J. Eng. Appl. Sci.*, vol. 6, no. 3, 2019, doi: 10.31873/ijeas/6.3.2019.30.

[11] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7304–7308.

[12] D. Chen and B. K.-W. Mak, "Multitask learning of deep neural networks for low-resource speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 7, pp. 1172–1183, 2015.

[13] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013, pp. 7319–7323, doi: 10.1109/ICASSP.2013.6639084.

[14] A. Kannan *et al.*, "Large-scale multilingual speech recognition with a streaming end-to-end model," *[C] Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2019-Septe, pp. 2130–2134, 2019, doi: 10.21437/Interspeech.2019-2858.

[15] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based End-to-End Models for Small-Footprint Keyword Spotting," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2018-Septe, pp. 2037–2041, Mar. 2018, doi: 10.21437/Interspeech.2018-1777.

[16] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings*, 2013, pp. 416–421, doi: 10.1109/ASRU.2013.6707766.

[17] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4955–4959.

[18] A. I. Garcia-Moral, R. Solera-Urena, C. Pelaez-Moreno, and F. Diaz-de-Maria, "Data balancing for efficient training of hybrid ANN/HMM automatic speech recognition systems," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 19, no. 3, pp. 468–481, 2010.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, vol. 2016-Decem, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[20] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Aug. 2016.

[21] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual Path Networks."

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," Nov. 2016.

[23] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks," *Proc. 23rd Int. Conf. Mach. Learn.*, pp. 369–376, 2006, doi: 10.1145/1143844.1143891.

[24] Y. Zhang and S. Id, "Speech Recognition Using Deep Learning Algorithms," 2013.

[25] M. Sun *et al.*, "Max-Pooling Loss Training of Long Short-Term Memory Networks for Small-Footprint Keyword Spotting," *2016 IEEE Work. Spok. Lang. Technol. SLT 2016 - Proc.*, pp. 474–480, May 2017, doi: 10.1109/SLT.2016.7846306.

[26] B. S. Michiel Hermans, "Training and Analysing Deep Recurrent Neural Networks," vol. 1. 100AD.

[27] W. Song and J. Cai, "End-to-End Deep Neural Network for Automatic Speech Recognition," 2015.

[28] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2017, doi: 10.1109/TNNLS.2016.2582924.

[29] Y. Wang and F. Tian, "Recurrent Residual Learning for Sequence Classification," 2016, pp. 938–943, doi: 10.18653/v1/d16-1093.

[30] Y. Zhuang, X. Chang, Y. Qian, and K. Yu, "Unrestricted vocabulary keyword spotting using LSTM-CTC," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2016, vol. 08-12-Sept, pp. 938–942, doi: 10.21437/Interspeech.2016-753.

[31] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks."

[32] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches."