Deep Speaker Verification Model for Low-Resource Languages and Vietnamese Dataset

Vi Thanh Dat
VNG corporation
datvt7@vng.com.vn

Pham Viet Thanh
Hanoi University of Science and Technology
thanh.pv.ds@gmail.com

Nguyen Thi Thu Trang*
Hanoi University of Science and Technology
trangntt@soict.hust.edu.vn

Abstract

Speaker verification is an essential task in speech processing with great authentication and surveillance applications. Large-scale datasets have hugely contributed to the success of neural networks for speaker verification. However, in low-resource languages, building such massive datasets is infeasible. This paper aims at proposing a speaker verification model for low-resource scenarios, with the baseline from Clova's H/ASP system in VoxSRC 2020. The proposed method adopts transfer learning to utilize the knowledge from the model trained with VoxCeleb, an English large-scale dataset. For network optimization, Stochastic Gradient Descent is employed instead of Adam because of its superior generalization. This work also proposes a novel marginal variant of Angular Prototypical (AP) Loss, i.e. Angular Margin Prototypical (AMP) Loss, which encourages a more discriminative embedding space. To experiment with the proposed model, we investigated building a public speaker verification dataset for Vietnamese. A processing pipeline is proposed to enhance the quality of the dataset. After being collected from sources, noisy speakers and noisy utterances are removed using self-similarity matrix analysis. Speakers with the same identity are then unified. The experimental results show that the proposed model achieves an Equal Error Rate (EER) of 3.1% which outperforms the baseline with 7.6% EER on the collected dataset.

*Corresponding author

1 Introduction

Speaker verification is a task which takes an unknown speech as input and determines whether the speech matches the claimed identity. With the developments of deep neural networks, speaker verification systems have gained huge advances and outperform traditional probabilistic systems.

The application of neural networks in speaker verification is to learn discriminative speaker embedding space. Commonly used architectures in pioneering works include TDNN-based models (Snyder et al., 2015) and networks architectures originated from computer vision researches such as VGG (Simonyan & Zisserman, 2014) and ResNet (He et al., 2016). Since speaker verification is a metric learning problem, the idea is to learn embedding features that have small intra-class distance and large inter-class distance. Several distinctive metric learning loss functions for speaker verification have been proposed, from the well-known Triplet Loss (Zhang et al., 2018) to Prototypical Loss (J. Wang et al., 2019) and Angular Prototypical Loss (Chung et al., 2020).

The success of speaker verification systems does not come only from the advancement of neural networks but is also attributed to the availability of large-scale speaker datasets. Good examples are VoxCeleb1 (Nagrani et al., 2017) and VoxCeleb2 (Chung et al., 2018) . These datasets together contribute more than a million utterances, spoken in English, from over 7,000 speakers. Furthermore, the two datasets have a complicated and well-designed data collection pipeline, thus can represent 'real world' situations. Using VoxCeleb2 as training data,

the author of (Chung et al., 2018) has achieved an Equal Error Rate (EER) of 3.95% on the test set of VoxCeleb1. Recently, a modified version of TDNN with multiple enhancements has been applied in (Desplanques et al., 2020) and obtained a new state-of-the-art result, an EER of 0.87% on the VoxCeleb1 test set.

However, it is difficult to achieve the same performance on low-resource languages. In the case of Vietnamese, there is only one publicly available dataset designed for experimenting with speaker recognition, which is from the ZaloAI Challenge 2020 [1]. The ZaloAI dataset consists of 8.7 hours of speech and 400 speakers. Compared to 2,000 hours of speech and 7,000 speakers from VoxCeleb, this dataset seems really small. Because of the scarcity of speaker data, there is no standardized benchmark for evaluating Vietnamese speaker verification models.

In this research, we propose a speaker verification model for low-resource languages. For the proposed model, we propose to use a modified version of Angular Prototypical Loss - Angular Margin Prototypical Loss - which further discriminates speakers in the embedding space. Additionally, transfer learning is applied to utilize the English pre-trained ResNet model from (Heo et al., 2020). To help improving speaker verification systems for Vietnamese, we build and publish VietSV, a new Vietnamese speaker verification dataset. The dataset is a combination of the ZaloAI dataset and other ASR datasets through a careful preprocessing and sampling pipeline.

The remaining of this paper is organized as follows. Our proposed model for low-resource languages is described in Section 2, with the related works given in Section 3. In Section 4, we discuss the process and result of building the new Vietnamese speaker verification dataset. Our experimental results are provided in Section 5. Finally, we draw conclusions in Section 6.

2 Proposed method

2.1 Baseline model

The architecture of the chosen baseline model is illustrated in Figure 1. This model is given as an unofficial baseline model for the VoxCeleb Speaker Recognition Challenge 2020 (Heo et al., 2020).

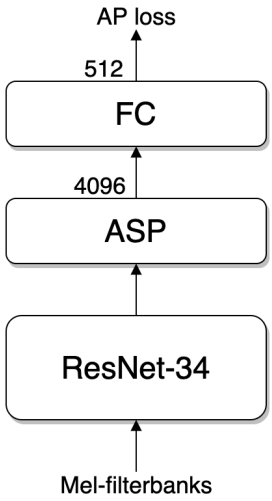


Figure 1: Baseline model architecture.

The model takes log Mel-filterbanks as input. The input then goes through a modified version of ResNet-34 to extract frame-level features. As input utterances have variable lengths, Attentive Statistics Pooling (ASP) (Okabe et al., 2018) is applied to aggregate frame-level features to obtain utterance-level features while keeping important information through an attention mechanism. Utterance-level features are then handled by fully connected layers (FC). Finally, the loss is computed using Angular Prototypical (AP) Loss with Adam optimization.

2.2 Proposed model

Figure 2 shows the overview of our proposed method. We utilize the pre-trained model of (Heo et al., 2020) with the architecture described in Section 2.1. Generally, using transfer learning can accelerate the training process and help the model achieves better outcomes when trained with low-resource data. Furthermore, as the experimenting speaker verification model is text-independent, choosing a model trained with a large dataset such as the English pre-trained model for fine-tuning will be a good option. To help the model further discriminate speakers in the embedding space, we use our proposed loss function - Angular Margin Prototypical Loss, which is discussed below. Additionally,

[1] https://challenge.zalo.ai

the backpropagation process is optimized with SGD for better generalization.

2.3 Angular Margin Prototypical Loss

2.3.1 Angular Prototypical Loss

As discussed in (Chung et al., 2020), AP Loss is a variant of Prototypical Loss (J. Wang et al., 2019) which optimize the embedding space by computing distances to the prototypes or centroids of every speaker.

Consider a mini-batch of N speakers, with each having M utterances, let $x_{i,j}$ be the embedding vector extracted from utterance j of speaker i, where $1 \leq i \leq N, 1 \leq j \leq M$. Suppose that we have $x_{i,M}$ as the query, the centroid of speaker i is calculated as:

$$c_i = \frac{1}{M-1} \sum_{m=1}^{M-1} x_{i,m} \tag{1}$$

The similarity score between the query and the centroid c_k of speaker k $(1 \leq k \leq N)$ is given in (2). Learnable scale w and bias b help to stabilize the convergence.

$$S_{i,k} = w \cdot cos(x_{i,M}, c_k) + b \tag{2}$$

Finally, the loss computed for a mini-batch is as follows:

$$L_{AP} = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{S_{i,i}}}{\sum_{k=1}^{N} e^{S_{i,k}}} \tag{3}$$

The softmax function is used here to push the embeddings of a speaker closer to his centroid and prevent the situation that those utterances have high similarity scores with other centroids. However, it has a problem that the distances between utterances of a speaker are still large and the distances to the decision boundaries created by the softmax function are small, as shown in Figure 3.

To deal with this problem, we introduce a new loss function, called Angular Margin Prototypical Loss (AMP), with two variants are AMP-cos and AMP-arc.

2.3.2 AMP-cos

In AMP-cos, a margin is added directly to the cosine similarity of 2 utterances. The similarity score in (2) is now replaced with:

$$S_{i,k} = \begin{cases} w \cdot (\cos\left(\theta_{x_{i,M},c_k}\right) - m) + b, & \text{if } i = k \\ w \cdot \cos\left(\theta_{x_{i,M},c_k}\right) + b, & \text{otherwise} \end{cases} \tag{4}$$

In (4), $\theta_{x_{i,M},c_k}$ is the angle between the query $x_{i,M}$ of speaker i and the centroid c_k of speaker k, w and b are learnable parameters and m is the additional margin. The objective function is the same as (3) but with the new similarity score.

2.3.3 AMP-arc

In the case of AMP-arc, an angular margin is added with the angle between 2 utterances:

$$S_{i,k} = \begin{cases} w \cdot \cos\left(\theta_{x_{i,M},c_k} + m\right) + b, & \text{if } i = k \\ w \cdot \cos\left(\theta_{x_{i,M},c_k}\right) + b, & \text{otherwise} \end{cases} \tag{5}$$

AMP-arc is more effective than AMP-cos because the angular margin exactly corresponds to geodesic distance, as discussed in (Deng et al., 2019). Adding angular margin improves intra-class compactness significantly as compared to AP loss. Figure 4 illustrates the decision boundaries created by AMP-arc.

3 Related works

Previous works have shown the efficiency of speaker verification models when using transfer learning and applying metric learning loss functions originated from the face recognition task. Sections below describe how these works have inspired us to come up with our proposed method.

3.1 Transfer learning

Transfer learning is widely applied in speaker verification and speaker recognition, usually when the target domain's data is scarce. Nidadavolu et al. (Nidadavolu et al., 2019) propose an adaptation technique by learning feature mapping function from low-resource target domain to source domain using CycleGAN in an unsupervised manner. Using Adversarial Discriminative Domain Adaptation and unlabeled target domain data, the work in (Xia et al., 2019) alleviates the domain mismatch problem in an English-Chinese cross-lingual speaker verification task. In the top-scoring submission for the text-independent task for SdSV challenge 2020, Thienpondt et al. propose Hard Prototype Mining loss to

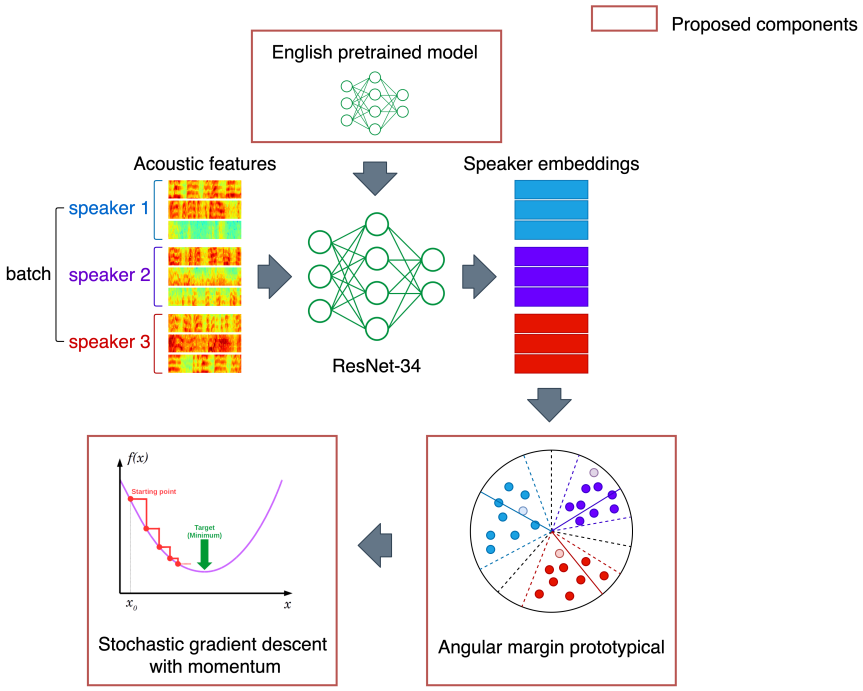


Figure 2: Proposed speaker verification model for low-resource languages with transfer learning from English pre-train model, SGD optimizer and Angular Margin Prototypical Loss.

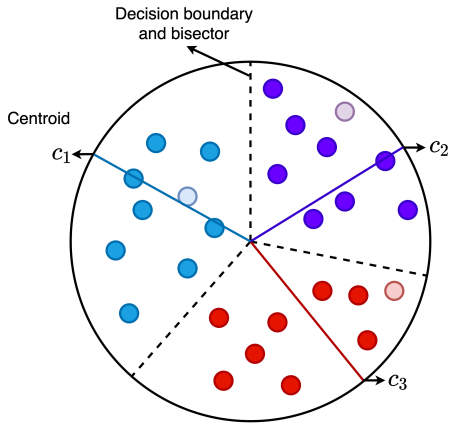


Figure 3: Embedding space learned with AP loss.

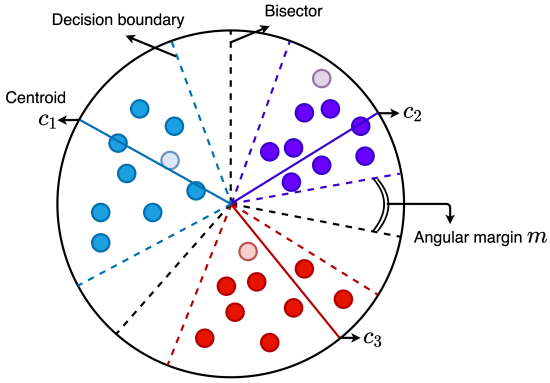


Figure 4: Embedding space learned with AMP-arc loss.

finetune a speaker verification model from English to Farsi language achieved 1.83% EER in the target language. Overall, transfer learning enables speaker verification in low-resource scenarios by leveraging knowledge learned from English or other rich-resource languages.

3.2 AMP Loss

Incorporating margin penalty in loss functions to enhances the inter-class separability and the intra-class compactness of embedding spaces is standard in face recognition. Margin is added in different ways to softmax function in A-softmax, CosFace and ArcFace (Liu et al., 2017; F. Wang et al., 2018;

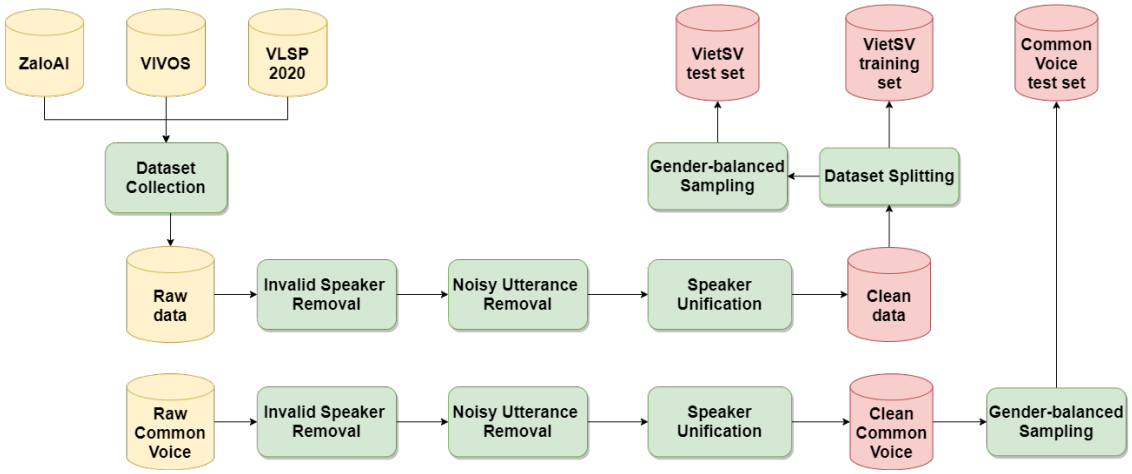


Figure 5: Overall processes of building VietSV and preparing Vietnamese Common Voice test set.

Deng et al., 2019). With its ease of implementation and good performance, ArcFace became popular and remains the state-of-the-art loss function for face recognition despite being published three years ago in 2018. Pair-based metric loss functions like contrastive loss (Hadsell et al., 2006), triplet loss (Hoffer & Ailon, 2015), lifted structure loss (Oh Song et al., 2016), histogram loss (Ustinova & Lempitsky, 2016), ... are usually designed with a margin penalty. Speaker verification with deep learning has successfully adopted these losses and surpassed the traditional i-vector in performance. Recently, inspired by ArcFace, Wei et al. (Wei et al., 2020) incorporate margin penalty in the batch-based loss GE2E which achieves clear performance improvement.

In this work, we incorporate margin into AP loss in both CosFace and ArcFace resembling manners. Although AP loss uses the same scoring function as GE2E loss, the centroid in AP loss is made from the same number of utterances. This stabilizes training and makes it possible to exactly mimic the test scenario during training, which has advantages over GE2E formation (Chung et al., 2020).

3.3 Network Optimization with SGD

Many literatures have shown that Adam tends to perform worse than SGD despite having faster training speed. (Zhou et al., 2020) has analyzed theoretically how SGD generalizes better than Adam in various tasks. Commonly, SGD is often used to get state of the art results for networks like ResNet (He et al., 2016), DenseNet (Huang et al., 2017), ResNeXt (Xie et al., 2017), SENet (Hu et al., 2018), ... While Adam is often used for large networks or complex systems such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), GANs (Goodfellow et al., 2014) due to its fast convergence and stability.

4 Vietnamese Dataset Building

To have a dataset for experimenting with the proposed model as well as publishing a good dataset for the research community, we have investigated building a dataset for Vietnamese, a low-resource language in this field of study.

4.1 Overall Process

We propose a procedure to build a speaker verification dataset for Vietnamese in particular, and for low-resource languages in general. The overall process is illustrated in Figure 5. The first step is the data collection phase, which combines speech datasets with speaker identities to obtain raw data. Then, clean data is acquired by pushing the raw data through several preprocessing substeps, including invalid speaker removal, noisy utterance removal and speaker unification. Lastly, the data is split into a training set and a test set, with the test set being gender-balanced sampled. Apart from the collected datasets, we also obtain Vietnamese speech from Common Voice [2]. As there are only 23 speakers and 253 utterances in the data, we will use it as an

[2]https://commonvoice.mozilla.org/en

out-domain test set. The same procedure excluding dataset splitting is applied to Common Voice data to obtain the Common Voice test set.

4.2 Dataset Collection

ZaloAI dataset consists of 400 speakers with 26.4 utterances per speaker on average. To extend the data used for the task, we select two more ASR datasets, which are VLSP 2020 [3] and VIVOS [4]. These two datasets are the only Vietnamese ASR datasets containing speaker information. There are 40 speakers who contributed to the VIVOS training set with each having 253.5 utterances on average. The validation set of VIVOS includes 19 speakers and 40 utterances per speaker on average. In the case of the VLSP 2020 dataset, the numbers of speakers and average utterances per speaker are 567 and 22.3, respectively.

Although the datasets are diverse in terms of gender, accent and age, there are flaws in the data such as mislabeled utterances and noisy audio. Furthermore, there are duplicate speakers in each dataset and in the combination of the datasets as well. These problems are handled using preprocessing techniques in the sections below.

4.3 Data Preprocessing

In the preprocessing techniques below, we analyze cosine similarity matrix on utterances to measure data consistency hence removing unwanted data. Given a set of n utterance embeddings $V = \{v_0, v_1, ..., v_{n-1}\}$, the cosine similarity matrix can be calculated in (6). Figure 6 shows the similarity matrix between utterances of a speaker.

$$S_{i,j} = cos(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\|\|v_j\|}, 0 \leq i, j \leq n \quad (6)$$

4.3.1 Invalid Speaker Removal

Some speakers have a large amount of noisy data, removing utterances manually can take a lot of time. Hence we use similarity matrix to assess whether to remove those speakers from the data. Figure 6 and Figure 7 show the cosine similarity matrices of a valid and invalid speaker, respectively.

[3] https://vlsp.org.vn/vlsp2020/eval/asr

[4] https://ailab.hcmus.edu.vn/vivos

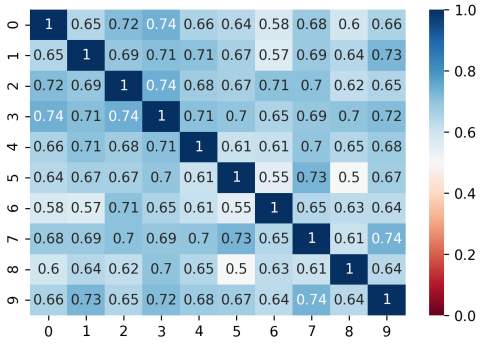


Figure 6: Cosine similarity matrix of a valid speaker with 10 utterances.

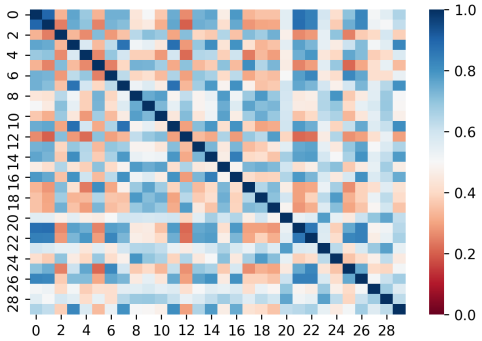


Figure 7: Cosine similarity matrix of an invalid speaker.

4.3.2 Noisy Utterance Removal

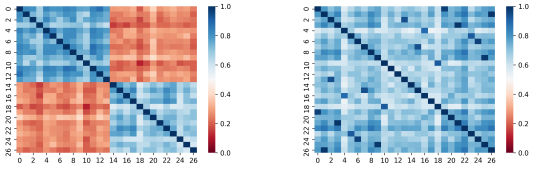
Noisy utterances can be spotted by using outlier detection (Yang et al., 2019). Let Q1 and Q3 be the first quartile and third quartile of the set of average similarity score $a_i = \frac{1}{n}\sum_{j=0,j\neq i}^{n-1} S_{i,j}$. By using interquartile range (IQR), the utterance valid range $[a_{min}, a_{max}]$ of set a can be determined in (7) and (8). Utterances having similarity scores a_i outside of the valid range $[a_{min}, a_{max}]$ will be marked and considered for removal.

$$a_{min} = Q1 - 1.5 * IQR; \; a_{max} = Q3 + 1.5 * IQR \quad (7)$$

$$IQR = Q3 - Q1 \quad (8)$$

4.3.3 Speaker Unification

As the four collected datasets are built independently, a pair of speakers from different datasets with the same identity can exist. We use similarity matrix to find these pairs. It is clear that in Figure 8a, 52-M-31 and 64-M-30 are different people as the similarity score between them is low. In contrast, there is a high probability that 64-M-30 and 636-M-30 in Figure 8b are the same speaker. Similarity matrices of pairs having average scores higher than 0.7 will be marked and considered for unification.



(a) Similarity matrix between 52-M-31 and 64-M-30.

(b) Similarity matrix between 64-M-30 and 636-M-30.

Figure 8: Similarity matrices of pairs of speakers.

Dataset	Speakers	Pairs of utterances
VietSV training set	1031	-
VietSV test set	59	48,148
Common Voice test set	23	12,192

Table 1: VietSV Datasets overview

Layer	Kernel size	Stride	Output shape
Conv1	$3 \times 3 \times 32$	1×1	$L \times 64 \times 32$
ResBlock1	$3 \times 3 \times 32$	1×1	$L \times 64 \times 32$
ResBlock2	$3 \times 3 \times 64$	2×2	$L/2 \times 32 \times 64$
ResBlock3	$3 \times 3 \times 128$	2×2	$L/4 \times 16 \times 128$
ResBlock4	$3 \times 3 \times 256$	2×2	$L/8 \times 8 \times 256$
Flatten	-	-	$L/8 \times 2048$
ASP	-	-	4096
Linear	512	-	512

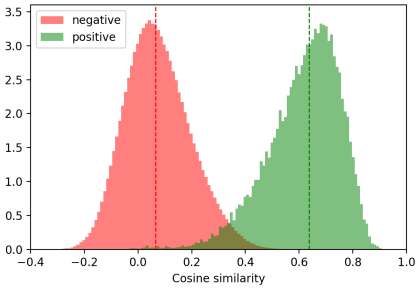
Table 2: Architecture of the performance-optimized ResNet-34

4.4 Dataset Construction

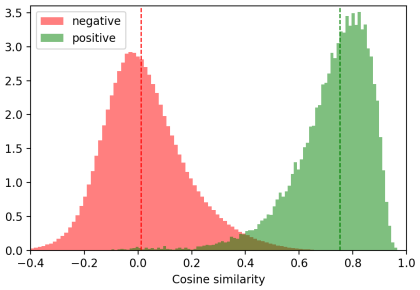
After removing 65 invalid speakers, 1,617 noisy utterances and unifying 84 duplicate speakers, we have a total of 1,113 speakers, of which 1090 speakers are from the collected dataset, the rest belongs to Common Voice data.

To construct the VietSV dataset, we perform dataset splitting step to sample speakers for the training set and the test set. As shown in Table 1, VietSV training set has 1,031 speakers. VietSV test set includes 59 speakers with 48,148 pairs of utterances generated from 1,626 utterances. In the set, we take 40 speakers randomly from ZaloAI with gender balance, the other 19 speakers are from the validation set of VIVOS. Common Voice test set is also gender-balanced sampled. VietSV will be publicly available to download at VLSP (http://vlsp.org.vn[5]).

5 Experiments



(a) Model trained with AP



(b) Model trained with AMP-arc (m=0.2)

Figure 9: Similarity score distribution of positive pairs and negative pairs. Dotted vertical line describes average similarity score.

5.1 Experimental setup

5.1.1 Speech feature extraction

We extract 64-dimensional log Mel-filterbank energies for each frame of width 25 ms and step 10 ms. The training segments are roughly 2s long speech

[5]Currently, VietSV can be downloaded at the link https://github.com/datvithanh/vietnamese-sv-dataset

Architecture	FT	Loss	m	Optimizer	VietSV EER	Common Voice EER
ECAPA (Thienpondt et al., 2020)	Y	HMP	-	Adam	4.017	-
ECAPA (Thienpondt et al., 2020)	Y	HMP	-	SGD	4.299	-
ResNet-34 pretrained	-	-	-	-	14.954	11.468
ResNet-34	N	AP	-	Adam	7.602	10.934
ResNet-34	Y	AP	-	Adam	5.446	8.002
ResNet-34	Y	AP	-	SGD	3.539	5.542
ResNet-34	Y	AMP-cos	0.1	SGD	3.319	5.100
ResNet-34	Y	AMP-cos	0.2	SGD	3.269	5.702
ResNet-34	Y	AMP-cos	0.3	SGD	3.232	6.687
ResNet-34	Y	AMP-cos	0.4	SGD	3.211	6.436
ResNet-34	Y	AMP-cos	0.5	SGD	3.331	8.102
ResNet-34	**Y**	**AMP-arc**	**0.1**	**SGD**	3.240	**4.789**
ResNet-34	**Y**	**AMP-arc**	**0.2**	**SGD**	**3.115**	5.391
ResNet-34	Y	AMP-arc	0.3	SGD	3.194	5.914
ResNet-34	Y	AMP-arc	0.4	SGD	3.298	6.757
ResNet-34	Y	AMP-arc	0.5	SGD	3.352	7.755

Table 3: Results on VietSV test set and VietSV Common Voice test set. The figures in bold represent the best results for each set. **FT**: fine-tuning.

signals, each segment generates a spectrogram of size 200×64. Similar to (Heo et al., 2020), mean and variance normalization is performed by applying instance normalization (Ulyanov et al., 2016) to the network input.

5.1.2 Neural network architecture

The architecture of the speaker discriminative DNN used in this work is illustrated in Table 2, which is similar to the architecture of the ResNet-34 used in (Heo et al., 2020). This variant of ResNet-34 has half of the channels and has the stride at Conv1 removed compared to the original ResNet-34 (He et al., 2016). After training, the 512-dimensional speaker embeddings are extracted from the Linear layer given the input features.

5.1.3 Implementation details

Experiments codes are implemented with Pytorch framework (Paszke et al., 2019). The models are trained using an NVIDIA V100 GPU with 16GB memory from Google's Colab Pro. One epoch is defined as a full pass through the speakers, each represented with a random 2s segment. We use an initial learning rate of 0.005, reduced by 25% every 50 epochs. The models are trained for 2000 epochs each. We use a mini-batch size of 200. Each model takes one day to train.

5.2 Evaluation protocol

The evaluation follows the standard protocol: extracting speaker embeddings and calculating their cosine similarity to determine speaker identities. We report the equal error rate (EER) for each speaker verification model.

5.3 Experimental results

Table 3 reports the experimental results.

We compare our models to our self-implemented version of the fine-tuning ECAPA models in (Thienpondt et al., 2020). The pre-trained ECAPA weights are available on Hugging Face (Wolf et al., 2019). The fine-tuned models are reported on VietSV with the author's Hard Prototype Mining Loss and the two optimizers Adam and SGD.

The results demonstrate that our AMP loss performs better than AP loss in in-domain scenarios. However, in the out-domain test - Test 2, AMP with higher values of m causes overfitting which significantly impairs the results on Test 2.

Figure 9 gives the score distributions of positive pairs and negative pairs for both AP loss and AMP loss. The verification performance is determined by the size of the overlap area between the negative pairs distribution and the positive pairs distribution. The angular margin effectively pushes the average scores away from each other.

The fine-tuned model trained with AMP-arc

($m = 1$) and optimized with SGD produces an EER of 3.115% relatively improved 11.98% and 59.02% upon the one trained with AP loss and baseline model for Vietnamese speaker verification.

6 Conclusions

In this paper, we have built the VietSV dataset for Vietnamese speaker verification. VietSV is collected from different sources and removed noises. The two test sets: the in-domain VietSV test set and the out-domain Common Voice test set are sampled from people with the same gender and accent to increase difficulty. We release the dataset to facilitate further research in Vietnamese speaker verification.

We have proposed a training system based on transfer learning to leverage learned knowledge from the large-scale English dataset - VoxCeleb. In our experiments, we found that switching from Adam optimizer to SGD optimizer substantially improves verification performance. We also propose a margin variant of the Angular Prototypical Loss - Angular Margin Prototypical Loss that outperforms the original training function by 11.98%. The overall system achieves 3.12% EER which outperforms the baseline H/ASP model with 7.60% EER.

References

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, *2*, 1735–1742.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Hoffer, E., & Ailon, N. (2015). Deep metric learning using triplet network. *International workshop on similarity-based pattern recognition*, 84–92.

Snyder, D., Garcia-Romero, D., & Povey, D. (2015). Time delay deep neural network-based universal background models for speaker recognition. *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 92–97.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Oh Song, H., Xiang, Y., Jegelka, S., & Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4004–4012.

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Ustinova, E., & Lempitsky, V. (2016). Learning deep embeddings with histogram loss. *arXiv preprint arXiv:1611.00822*.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220.

Nagrani, A., Chung, J. S., & Zisserman, A. (2017). Voxceleb: A large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Chung, J. S., Nagrani, A., & Zisserman, A. (2018). Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the*

IEEE conference on computer vision and pattern recognition, 7132–7141.

Okabe, K., Koshinaka, T., & Shinoda, K. (2018). Attentive statistics pooling for deep speaker embedding. *arXiv preprint arXiv:1803.10963*.

Wang, F., Cheng, J., Liu, W., & Liu, H. (2018). Additive margin softmax for face verification. *IEEE Signal Processing Letters*, *25*(7), 926–930.

Zhang, C., Koishida, K., & Hansen, J. H. (2018). Text-independent speaker verification based on triplet convolutional neural network embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(9), 1633–1644.

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4690–4699.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Nidadavolu, P. S., Kataria, S., Villalba, J., & Dehak, N. (2019). Low-resource domain adaptation for speaker recognition using cycle-gans. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 710–717.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Wang, J., Wang, K.-C., Law, M. T., Rudzicz, F., & Brudno, M. (2019). Centroid-based deep metric learning for speaker recognition. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3652–3656.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xia, W., Huang, J., & Hansen, J. H. (2019). Cross-lingual text-independent speaker verification using unsupervised adversarial discriminative domain adaptation. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5816–5820.

Yang, J., Rahardja, S., & Fränti, P. (2019). Outlier detection: How to threshold outlier scores? *Proceedings of the international conference on artificial intelligence, information processing and cloud computing*, 1–6.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chung, J. S., Huh, J., Mun, S., Lee, M., Heo, H. S., Choe, S., Ham, C., Jung, S., Lee, B.-J., & Han, I. (2020). In defence of metric learning for speaker recognition. *arXiv preprint arXiv:2003.11982*.

Desplanques, B., Thienpondt, J., & Demuynck, K. (2020). Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*.

Heo, H. S., Lee, B.-J., Huh, J., & Chung, J. S. (2020). Clova baseline system for the voxceleb speaker recognition challenge 2020. *arXiv preprint arXiv:2009.14153*.

Thienpondt, J., Desplanques, B., & Demuynck, K. (2020). Cross-lingual speaker verification with domain-balanced hard prototype mining and language-dependent score normalization. *arXiv preprint arXiv:2007.07689*.

Wei, Y., Du, J., & Liu, H. (2020). Angular margin centroid loss for text-independent speaker recognition. *Proc. Interspeech 2020*, 3820–3824.

Zhou, P., Feng, J., Ma, C., Xiong, C., Hoi, S., et al. (2020). Towards theoretically understanding why sgd generalizes better than adam in deep learning. *arXiv preprint arXiv:2010.05627*.