EasyChair Preprint

№ 10680

Comparison of Different Neural Network Architectures for Spoken Language Identification

Tala Bazazo, Mohammad Zeineldeen, Christian Plahl,
Ralf Schlüter and Hermann Ney

August 7, 2023

Comparison of Different Neural Network Architectures for Spoken Language Identification

Tala Bazazo[1,2], Mohammad Zeineldeen[1], Christian Plahl[2], Ralf Schlüter[1], Hermann Ney[1]

[1]Human Language Technology and Pattern Recognition, RWTH Aachen University, Germany
[2]eBay, Aachen, Germany
Email: tala.bazazo@rwth-aachen.de, {zeineldeen, schlueter, ney}@cs.rwth-aachen.de, cplahl@ebay.com

Abstract

This paper compares different neural network based architectures on the spoken language identification task. To our best knowledge such a comparison of different models on the same dataset and the same set of languages does not yet exist. We incorporate 7 different models which include the latest architectures: a spectral images based Resnet model, a Convolutional Neural Network, a Bi-directional Long Short-Term Memory, a Convolutional Recurrent Neural Network, Wav2Vec 2.0, a transformer and a conformer. We also tackle audio with background noise and music by training on data with similar accoustics. We finally also show that our models generalize well on third-party data.

1 Introduction & Related Work

Spoken language identification (LID) is the process of classifying the language spoken in a speech recording [1]. In fact, it has been a research topic for a long time.

Up until the late 90s, the main focus for LID has been the feature extraction, where language dependent features have been extracted to classify the language. [1] summarizes the typical approaches which include: spectral-similarity, prosody, phone, multilingual speech units, word Level and continuous speech recognition based approaches. Spectral-similarity based approaches compute a set of short-term spectra from training samples and compare them with those of testing by calculating the distance between them. Prosody-based approaches use features based on pitch estimates alone, while phone-recognition based approaches hypothesize exactly which phones are being spoken as a function of time and determines the language based on the statistics of that phone sequence. Multilingual speech units based approaches derive either a mixture of language dependent and independent phones or by deriving tokens automatically from training data. Word level based approaches handle an incoming utterance by processing it through parallel language-dependent phone recognizers. This is known as a bottom-up approach where phones are recognized first, followed by words, and the language afterwards. Finally, continuous speech recognition approaches create one speech recognizer per language during training, and then during testing, each of these recognizers is run in parallel. The recognizer with the highest likelihood is selected and its corresponding language used for training is the winning language. Later on, identity vectors [2] have appeared during speaker verification systems and have showed great success [3] including LID.

In the last years, neural networks achieved remarkable results in AI and found their way into the LID domain as well. Neural networks are used to do the feature extraction as well as the whole classification tasks. For example, in [4] and [5], the LID task is treated as an image classification task where mel-spectrograms [6] are extracted from the audio, transformed into images (either gray scale or RGB) and then fed to a model for the classification. On the other hand, other papers tackle the LID task in a similar fashion as automatic speech recognition (ASR). These systems use acoustic features such as Mel Frequency Cepstral Coefficients (MFCCs) or log-mel spectrograms [6] as input to the neural network architecture. In [7], the authors extract MFCC features from the audio files and feed those as input to 3 different neural network architectures: convolutional neural network (CNN), convolutional recurrent neural network (CRNN) and CRNN with attention to classify 13 Indian languages. Similarly, [8] uses MFCC features [9] as input to a long short-term memory (LSTM) architecture. Other works rely on i-vectors [10] as input to the neural network [11].

Recently, transformer [12] based architectures became popular in many fields such as ASR. In [13], a conformer [14] has been used in a joint task between ASR and LID. A decoder for each target language has been added for feature extraction followed by a ResNet architecture with 1D-convolutional layers for LID. Wav2vec2.0 [15] is also one of the latest models based on transformers to be used for ASR and audio classification tasks. [16] has used a pre-trained Wav2Vec 2.0 for speaker verification and LID by building on the features created by Wav2Vec an average pooling layer and a fully connected layer that is then fine-tuned on the actual recognition task.

Although neural networks have been widely used for LID, a detailed comparison of these approaches on a publicly-available data set with the same set of languages has not been done yet to our best knowledge. This is important because with all the different models out there, it is difficult to compare the models and tell which one of them works best when different datasets and languages are used. In this work, we give an overview of the current state-of-the-art neural network architectures for LID. We compare and evaluate the performance of 7 different models on the LID task for 6 European languages on two public available data sets. These models include: a spectral images based Resnet [17] model, a CNN [18], a Bi-directional LSTM (BLSTM)[19], a CRNN [20], the self-supervised Wav2Vec 2.0 [15], a transformer [12] and a conformer [14]. We also carry out an investigation on how noise and background music can be tackled by incorporating corpora with different acoustic conditions, which was also not done before on all those different models yet to our best knowledge.

2 Model Architectures

In this work, seven different model architectures are compared: a Spectral Images based model, a CNN, a BLSTM, a CRNN, a Wav2Vec, a transformer and a conformer architecture. We present the model architectures and the corresponding features which are mostly reported in literature and which gives us the best results.

2.1 Spectral Images Based

This model follows the work in [5], where LID is treated as an image classification task. The 40-dimensional mel-spectrograms extracted from the raw audio are scaled to RGB images resulting in the shape (40, T, 3) with T being the number of frames. These features are referred to as spectral images. A pre-trained Resnet50 with 2 linear layers of size 2048 followed by a softmax layer is fine-tuned on the spoken LID task by feeding in the spectral images.

2.2 Convolutional Neural Network

23-dimensional MFCCs from the raw audio are z-normalized per sequence and are fed into a TDNN architecture following [21], which is a 1D CNN with dilations. Table 1 shows the TDNN architecture in detail.

Layer	Layer context	Total context	In x out
Frame1	[t-2, t+2]	5	5Fx512
Frame2	{t-2, t, t+2}	9	1536x512
Frame3	{t-3, t, t+3}	15	1536x512
Frame4	{t}	15	512x512
Frame5	{t}	15	512x1500
Stats pooling	[0,T}	T	1500Tx3000
Segment6	{0}	T	3000x512
Segment7	{0}	T	512x512
Softmax	{0}	T	512xL

Table 1: TDNN architecture from [21]

2.3 Bi-directional Long Short-Term Memory

40-dimensional MFCCs which are z-normalized per sequence are fed as input to a BLSTM with 2 layers and 512 units each. The output layer combines the forward and backward pass into a single vector and then feeds it to a softmax layer. The vector is obtained by:

$$o = [\overrightarrow{o}, \overleftarrow{o}] \tag{1}$$

where:

$$\overrightarrow{o} = \frac{\sum_{t=1}^{T} \overrightarrow{LSTM}(h_t)}{T} \tag{2}$$

$$\overleftarrow{o} = \frac{\sum_{t=T}^{1} \overleftarrow{LSTM}(h_t)}{T} \tag{3}$$

and h is the hidden unit and t is the frame index

2.4 Convolutional Recurrent Neural Network

This model combines the CNN and BLSTM models mentioned in the previous sections. First, the 23-dimensional z-normalized MFCCs are fed into the CNN component followed by the BLSTM architecture.

2.5 Wav2Vec 2.0

Wav2Vec 2.0 is a self-supervised model, pre-trained on unlabeled data and can be fine-tuned on labeled data afterwards. We first review the pre-training of the Wav2Vec 2.0 [15]. Then we introduce how to apply the pre-trained model to downstream tasks. As shown in Figure 1 the Wav2Vec 2.0 model consists of two parts: a front-end CNN network followed by a Transformer model. Wav2Vec 2.0 takes the raw audio sequence x_1^T as input. Then, latent speech representations z_1^T are obtained followed by contextualized representations c_1^T. The latent speech representations are also quantized to q_1^T. A masking scheme is introduced here where part of the transformer's input is masked and the aim is to guess the masked latent feature vector representation z_t.

The loss used for self-supervised pre-training is defined as follows:

$$L = \frac{1}{|T_{masked}|} \sum_{t \in T_{masked}} -log \frac{s(q_t, c_t)}{\sum_{\tilde{q} \in \tilde{Q}_t} s(\tilde{q}_t, c_t)} \tag{4}$$

where T_{masked} are the masked samples and

$$s(q_t, c_t) = exp(\frac{1}{T} \cdot \frac{q_1^T c_t}{\|q_t\| \cdot \|c_t\|}) \tag{5}$$

with $\tilde{Q}_t$ containing the true sample q_t and N-1 negative samples and $s(q_t, c_t)$ is a similarity function.

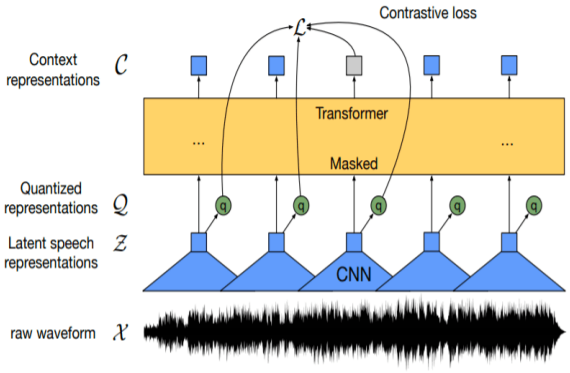

Figure 1: Wav2Vec 2.0 Architecture taken from [15]

Another model based on Wav2Vec 2.0 is the XLSR-53 [22] model. The model is pre-trained on 53 different languages (very inclusive languages from all over the world and from very different kind of families) and overall 56k hours from MLS [23], CommonVoice [24] and BABEL [25]. Detailed description of both models are given in the original papers [15] and [22].

In this work, 2 different setups are implemented for the Wav2Vec 2.0 model:

- XLSR-finetune: fine-tuning of the Wav2Vec 2.0 XLSR-53 [22] on the LID task using the raw audio as input. The Wav2Vec 2.0 XLSR-53 model is modified by adding two additional linear layers of size 1024 and a softmax layer at the end.
- Wav2Vec-pre-train:
  - Pre-training of the Wav2Vec 2.0 model on the LID data only. Due to the size of the training data, the model is reduced in the size and contains only 4 attention heads, 8 layers, 256 hidden dimension, and 1024 intermediate dimension. Again, 2 linear layers and a softmax layer are added.
  - Fine-tuning on the LID task.

2.6 Transformer

40-dimensional MFCCs which are z-normalized per sequence are fed as input to a Transformer architecture with only an encoder of 10 layers, 4 attention heads, 256 model dimension and 1024 hidden dimension. No positional encoding was used. The frames obtained after the encoder layer are pooled by their mean to a single vector. A linear layer was added before the encoder with size 256 and one after with size 1024 followed by a softmax.

2.7 Conformer

Similar to the Transformer, same feature extraction and model architecture are used with the exception of adding intermediate depth-wise convolutions inside the conformer block of kernel size 31.

3 Experiments and Results

3.1 Training and Testing Corpora

In this work, the following 6 European languages have been chosen for language identification: English, German, French, Spanish, Italian and Russian. The training and testing data are taken from two public data sets which allows readers to easily reproduce the results obtained in this work [1]. The data sets cover different acoustic conditions, e.g. clean speech and speech with background noise and music. The two public data sets are VoxForge [26] and Voxlingua107 [27] and are introduced briefly below.

3.1.1 VoxForge

VoxForge [26] includes material from popular classic fiction, poetry, plays and religious texts. It mainly consists of audio utterances with clean speech and are on the shorter side. Out of available languages the 6 languages mentioned above are chosen.

3.1.2 Voxlingua107

On the other hand, Voxlingua107 [27] includes speech segments that are automatically extracted from Youtube videos. The audio consists of noise and music in the background and is a more realistic speech collection. The videos are segmented to shorter segments with around 10 seconds of speech. The full training set consists of 107 languages but we only select the same 6 languages as for VoxForge [2].

3.1.3 Corpus Splits

Set	Characteristic	English	German	Russian	Italian	Spanish	French
Train	Avg Length [s]	5	5	8	7	8	6
	# of Speakers	291	90	193	152	280	195
	# of Hours [h]	13.6	13.6	13.6	13.6	13.6	13.6
Validation	Avg Length [s]	5	5	5	7	7.5	6.5
	# of Speakers	27	7	7	16	10	8
	# of Hours [h]	1	1	1	1	1	1
Test	Avg Length [s]	5	5	3	7	7	6
	# of Speakers	41	7	8	15	18	9
	# of Hours [h]	3	3	3	3	3	3

Table 2: Data Statistics on VoxForge

Set	Characteristic	English	German	Russian	Italian	Spanish	French
Train	Avg Length [s]	10.5	11	10.5	9	10	9
	# of Speakers	199	194	126	185	207	139
	# of Hours [h]	13.6	13.6	13.6	13.6	13.6	13.6
Validation	Avg Length [s]	10.5	11	10	8	10	9
	# of Speakers	22	20	10	18	7	8
	# of Hours [h]	1	1	1	1	1	1
Test	Avg Length [s]	11	10.5	10	9.5	10.5	9.5
	# of Speakers	62	52	33	44	34	27
	# of Hours [h]	3	3	3	3	3	3

Table 3: Data Statistics on Voxlingua6

There is no pre-defined split into train, validation and test sets available. Therefore, we create these sets for each data source with the following constrains: no speaker overlap in the different sets, different accents and gender in all sets and the same prior probabilities for each language. Overall, each data set and each language consists of 7000 utterances for training, 500 utterances for validation and 1500 utterances for testing. Tables 2 and 3 below show the data statistics on our data used.

3.1.4 Voxlingua107 Dev Set

Voxlingua107 provides a dedicated development set, which contains a few samples for most of the 107 languages but is not balanced as shown in Table 4. The utterances within this set contain audio with more background noise. We used the data set for more insights of the models' performance.

Characteristic	English	German	Russian	Italian	Spanish	French
Avg Length [s]	11.5	10	11.5	8	10.5	10
# of Speakers	80	82	29	49	55	100
# of Samples	80	82	29	49	55	100

Table 4: Data Statistics on the Voxlingua Dev set

3.2 Experimental Setup

3 different dataset settings are used for the model training. Each of the models are trained only on voxForge and VoxLingua6 as well as on both sets. We add cross data set evaluation to see the performance gain when switching from a single data set to both sets. The normal training includes the whole utterance, but an alternative model is trained on a reduced audio length. In such a case the latency is low and language dependent actions can be triggered earlier. The reduced audio lengths chosen are 3 and 5 seconds. The average length for the different data sets are shown in Table 2 and Table 3.

Finally, a data augmentation scheme called SpecAugment [28] is included for all models. All our models are implemented in the pyTorch framework [29] and the Wav2Vec models are from Hugging Face [30].

3.3 Training and Hyper-parameters

For the spectral images based model, stochastic gradient descent with a maximum learning rate of 0.1 and 0.9 momentum are used. 1-Cycle learning rate scheduler [31] is used with cycle length of 8, which anneals the learning rate from an initial learning rate to a maximum learning rate of 0.1 and then back to some minimum learning rate much lower than the initial one.

For the CNN, BLSTM and CRNN models, AdamW optimizer [32] and constant learning rate of 0.001 are used as well as dropout [33] of 0.3 right after every fully connected layer.

For XLSR-finetune, AdamW optimizer and a linear learning rate scheduler with 500 warmup steps is used, which linearly increases the learning rate to a maximum of 3e-4 and then linearly decreases it again.

For Wav2Vec-pre-train, pre-traning is done with the contrastive loss, AdamW optimizer and a linear learning rate scheduler with 9,000 warm up steps and a maximum learning rate of 5e-5 are used. For fine-tuning, AdamW optimizer and a linear learning rate scheduler with 5000 warm up steps and a maximum learning rate of of 3e-4.

For Transformer and Conformer, AdamW optimizer and a linear scheduler with a wamup of 1/3 the total number of steps and a maximum learning rate of 0.001 are used.

All models use the cross entropy loss for training except for the pre-training of Wav2Vec-pre-train and are trained for 32 epochs except the spectral images based.

3.4 Results and Discussion

Table 3.4 summarizes the size of each model trained. Besides the XLSR-finetune model, all models are designed to

[1] Link to the data splits will be provided later.

[2] We call this dataset Voxlingua6.

Model	Spec-Augment	VoxForge			Voxlingua6			VoxForge+Voxlingua6					
								VoxForge			Voxlingua6		
		Input Length [s]											
		3	5	full	3	5	full	3	5	full	3	5	full
Spectral Images	no	5.01	3.46	2.92	16.03	9.12	6.10	4.78	2.98	2.12	14.15	7.68	4.57
BLSTM	no	9.86	4.35	4.97	21.41	14.32	9.12	7.99	4.72	3.74	19.88	11.95	8.73
	yes	6.45	3.97	3.66	19.85	11.24	8.01	6.14	3.10	2.79	16.78	11.12	7.31
CNN	no	6.47	4.22	3.15	20.78	10.16	7.02	7.82	3.23	2.91	19.13	11.01	6.91
	yes	6.03	3.87	3.05	17.25	9.39	7.54	5.89	2.76	2.11	16.45	9.93	5.23
CRNN	no	5.46	3.97	3.92	19.64	9.91	6.96	5.03	3.04	2.68	14.17	7.84	4.21
	yes	4.07	1.96	1.54	16.88	8.78	6.43	4.11	1.45	1.01	13.83	6.76	3.97
XLSR-finetune	yes	2.97	1.32	1.12	6.43	3.68	2.31	2.12	0.98	0.33	4.78	2.95	1.47
Wav2Vec-pre-train	yes	10.53	7.24	3.81	24.05	14.92	8.65	9.21	6.32	4.22	22.04	12.11	7.29
Transformer	yes	5.13	3.72	2.88	16.91	10.11	6.77	4.38	2.93	2.09	14.09	7.02	4.00
Conformer	yes	3.24	1.89	1.31	9.22	5.56	4.41	2.82	1.26	0.93	6.67	4.63	2.69

Table 5: Error Rates in [%] for all the models trained on the different datasets

have a comparable number of parameters.

Model	# of Parameters [million]
Spectral Images	23.5
BLSTM	8.5
CNN	6.0
CRNN	19.5
XLSR-finetune	316.0
Wav2Vec-pre-train	11.7
Transformer	8.0
Conformer	15.5

Table 6: Total Number of Parameters of Each Model

3.4.1 VoxForge and Voxlingua

Table 3.3 summarizes the comparison of the different models trained on the different data sets and the different audio length. The table shows that all models perform better on VoxForge than Voxlingua6 due to the nois accoustics found in the latter. XLSR-finetune model outperforms all the other models. This is mainly due to the large model size (316 million parameters) and the very long number of hours that the model was pre-trained on. Therefore, in order to be fair, we have to look at the table while ignoring that model and compare the other models. We therefore observe that the Conformer model gives the best results among all models on all datasets: VoxForge, Voxlingua6 and the combination of both. The results are also very impressive and close to the XLSR-finetune model knowing the former is only trained on our relatively small datasets and number of parameters. The CRNN model comes in second place after adding SpecAugment followed by the Transformer. The spectral images based model also shows some promising results. This can be explained by the fact that mel-spectograms converted to images can be more robust than MFCCs with capturing language features. Furthermore, we also deduce that training on noisy data can help tackle noise and background music and that is by training on Voxlingua6. Training on both datasets also boosts the performance on each of the datasets and makes the model more robust. This can be seen in all models that when trained on the combination of the datasets, the model's performance is boosted on the individual datasets.

3.4.2 Voxlingua107 dev set

Table 3.4.2 shows the results on the dedicated Voxlingua107 dev set, where the model has been trained on the full utterance including SpecAugment. Ignoring the XLSR-finetune results for the same reason mentioned in the previous section, the Conformer model also gives the best results with 5.27% error rate. The CNN model surprisingly comes in the second place with 7.18% error rate followed by Transformer and BLSTM. Other models show some promising results as well. The models trained in this work seem to be robust and can generalize well on unseen harder data.

Model	Error Rate [%]
Spectral Images	12.44
BLSTM	11.24
CNN	7.18
CRNN	11.35
XLSR-finetune	2.09
Wav2Vec-pre-train	11.26
Transformer	8.56
Conformer	5.27

Table 7: Error Rates [%] of all models tested on Voxlingua Dev set

4 Conclusions

In this work, we implemented different neural network architectures and investigated the different approaches that can be used for LID. We also utilized the latest state-of-the-art (SOTA) models such as Wav2Vec 2.0 [15], Transformer [12] and Conformer [14] for LID. This comparison including the latest models has not been done yet to our knowledge. We also tried to tackle data with background noise and music by training on acoustically matching data. We concluded that the Conformer model was the best performing model and the SOTA for not only speech recognition but also LID. We also observed that simple neural networks with very few parameters are very good for LID. A larger model does not necessarily mean a better performance as seen with the Conformer, Transformer and CNN models, and pre-training on long number of hours helps a lot as seen with the XLSR-finetune model and with our training on both datasets. In the future, several things can still be explored such as adding more languages from different roots. It would be also interesting to see if adding any phonetic information to the models as done in [34] can help boost the performance even more.

References

[1] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Communication*, vol. 35, no. 1, pp. 115–124, 2001. MIST.

[2] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[3] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," pp. 857–860, 08 2011.

[4] C. Korkut, A. Haznedaroglu, and L. Arslan, "Comparison of deep learning methods for spoken language identification," in *Speech and Computer* (A. Karpov and R. Potapova, eds.), (Cham), pp. 223–231, Springer International Publishing, 2020.

[5] S. Revay and M. Teschke, "Multiclass language identification using deep learning on spectral images of audio signals," 2019.

[6] Mlearnere, "Learning from audio: The mel scale, mel spectrograms, and mel frequency cepstral coefficients," Apr 2021.

[7] A. Mandal, S. Pal, I. Dutta, M. Bhattacharya, and S. K. Naskar, "Is attention always needed? a case study on language identification from speech," *SSRN Electronic Journal*, 2022.

[8] R. Zazo, A. Lozano-Diez, J. Gonzalez-Dominguez, D. T. Toledano, and J. Gonzalez-Rodriguez, "Language identification in short utterances using long short-term memory (lstm) recurrent neural networks," *PLOS ONE*, vol. 11, pp. 1–17, 01 2016.

[9] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of mfcc.," *J. Comput. Sci. Technol.*, vol. 16, pp. 582–589, 11 2001.

[10] V. Vestman, K. A. Lee, and T. H. Kinnunen, "Neural i-vectors," 2020.

[11] W. Cai, D. Cai, S. Huang, and M. Li, "Utterance-level end-to-end language identification using attention-based cnn-blstm," 2019.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[13] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020* (H. Meng, B. Xu, and T. F. Zheng, eds.), pp. 5036–5040, ISCA, 2020.

[14] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[15] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.

[16] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," 2020.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV, USA), pp. 770–778, June 2016.

[18] S. Saha, "A comprehensive guide to convolutional neural networks the eli5 way," Nov 2022.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[20] X. Fu, E. Ch'ng, U. Aickelin, and S. See, "Crnn: A joint neural network for redundancy detection," 2017.

[21] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken Language Recognition using X-vectors," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)*, pp. 105–111, 2018.

[22] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," 2020.

[23] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *ArXiv*, vol. abs/2012.03411, 2020.

[24] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," 2019.

[25] A. R. Punnakkal, A. Chandrasekaran, N. Athanasiou, A. Quiros-Ramirez, and M. J. Black, "BABEL: Bodies, action and behavior with english labels," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, (Piscataway, NJ), pp. 722–731, IEEE, June 2021.

[26] Voxforge.org, "Free speech... recognition (linux, windows and mac) - voxforge.org." `http://www.voxforge.org/`. accessed 06/25/2014.

[27] J. Valk and T. Alumäe, "VoxLingua107: a dataset for spoken language recognition," in *Proc. IEEE SLT Workshop*, 2021.

[28] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*, ISCA, sep 2019.

[29] "Pytorch," 2016.

[30] "Hugging face," 2016.

[31] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," 2017.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017.

[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012.

[34] H. Liu, L. P. G. Perera, A. W. H. Khong, S. J. Styles, and S. Khudanpur, "Pho-lid: A unified model incorporating acoustic-phonetic and phonotactic information for language identification," 2022.