Tone Detection in Vietnamese Speech with CNNs

Ramon Casas

Introduction

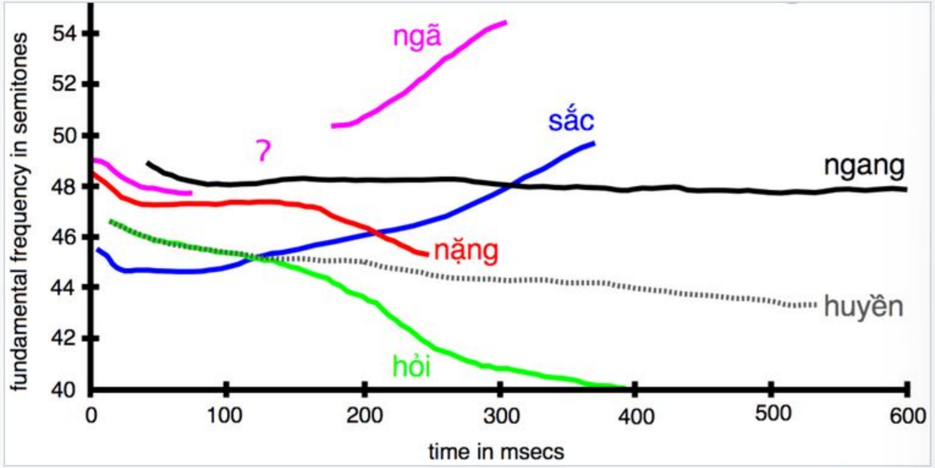
Vietnamese is a tonal language (6 tones) where meaning of words change depending on pitch contour.

Goal: Automatically classify spoken Vietnamese syllables (in isolation) by tone.

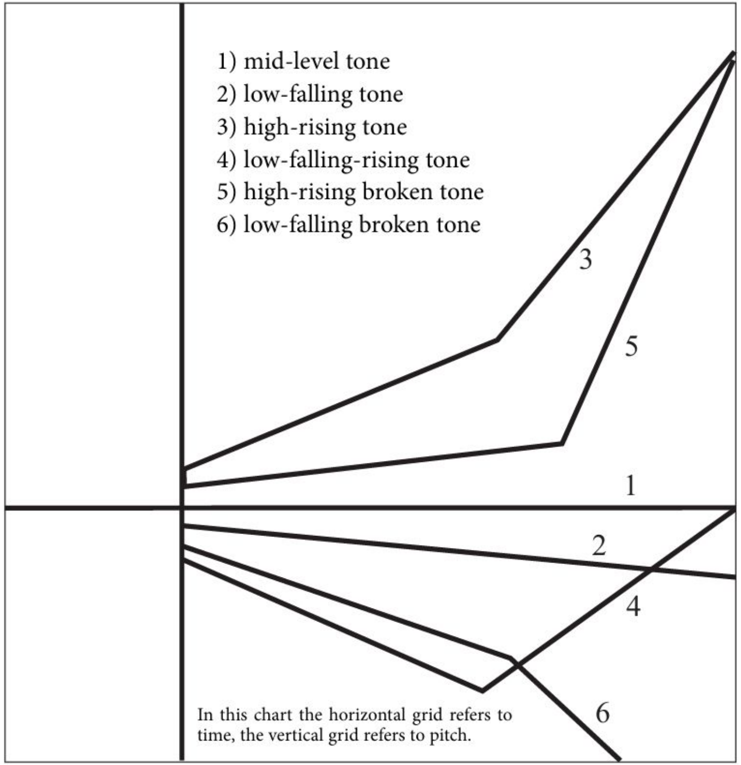
Motivation:

- Help Automatic Speech Recognition tasks.
- Use it as a learning Tool.

Introduction



Hanoi tones as uttered by a female speaker in isolation. From Nguyễn & Edmondson (1998)



State of the Art

- **Tone Classification** of Mandarin using CNNs: ToneNet (Gao et al. 2019)
- Vietnamese **Voice Classification** based on DL (Bui Thanh Hung, 2020)
- End-to-End Mandarin Tone Classification with **short term context information** (Tang and Li, 2021)
- Mandarin **Tone Modelling** Using **RNNs** (Huang, Hu and Xu, 2017)

State of the Art

ToneNet: A CNN Model of Tone Classification of Mandarin Chinese

Qiang Gao, Shutao Sun, Yaping Yang*

School of Computer and Cyberspace Security, Communication University of China
Beijing, China
qiangao, stsun, yyp_berry@cuc.edu.cn

MANDARIN TONE MODELING USING RECURRENT NEURAL NETWORKS

Hao Huang, Ying Hu*
School of Information Science and Engineering
Xinjiang University
Urumqi, China, 830046

Haihua Xu
Temasek Laboratories
Nanyang Technological University
Singapore, 637553

End-to-End Mandarin Tone Classification with Short Term Context Information

Jiyang Tang* and Ming Li*
* Data Science Research Center, Duke Kunshan University, Kunshan, China
E-mail: {jiyang.tang, ming.li369}@dukekunshan.edu.cn

IJMLNCE JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2020v04i04.004

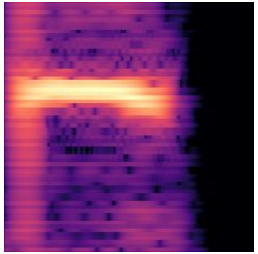
Vietnamese Voice Classification based on Deep Learning Approach

[a]Bui Thanh Hung

[a]*Faculty of Information Technology,Ton Duc Thang University, 19 Nguyen Huu Tho Street, Tan Phong Ward, District 7, Ho Chi Minh City, Vietnam, buithanhhung@tdtu.edu.vn*

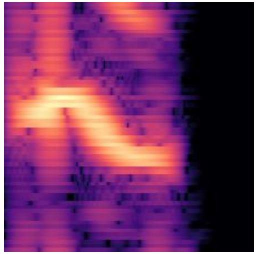
Methodology: Data Analysis and Preprocessing

Synthetic Data Generation

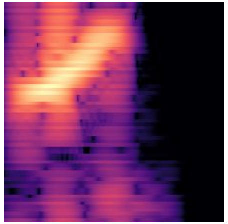


Semiautomatic Extraction of Syllables

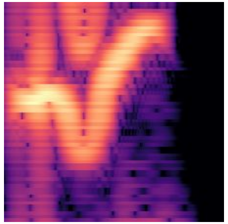
Downloading data from pronunciation websites



MEL Spectrogram Extraction
librosa library

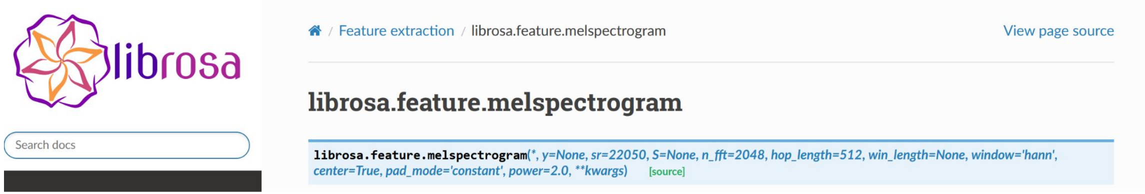


.npy files

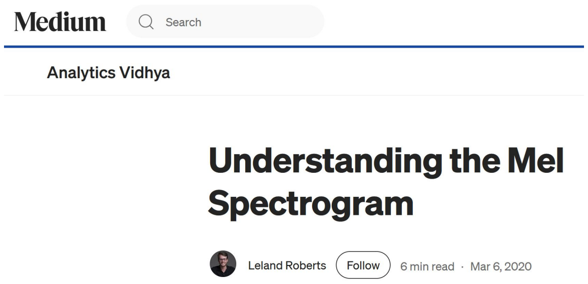


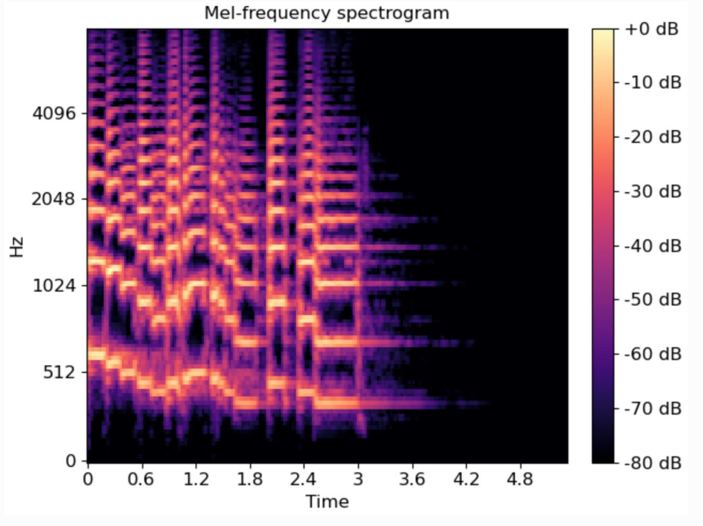
.png files

Methodology: Data Analysis and Preprocessing

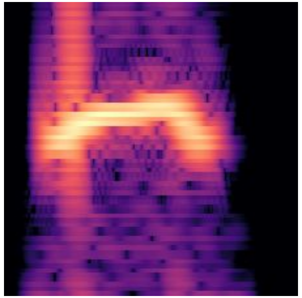


MEL
Spectrogram
Extraction
librosa library

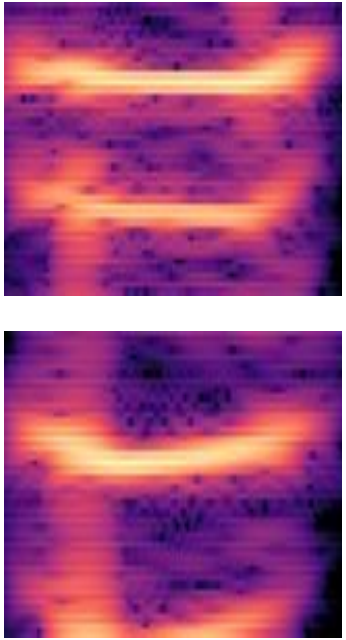




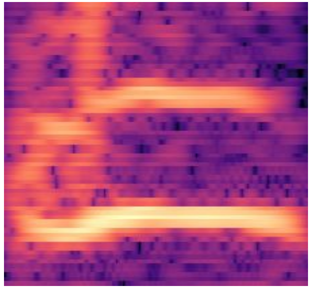
Building the datasets: Synthetic Data



Google TTS

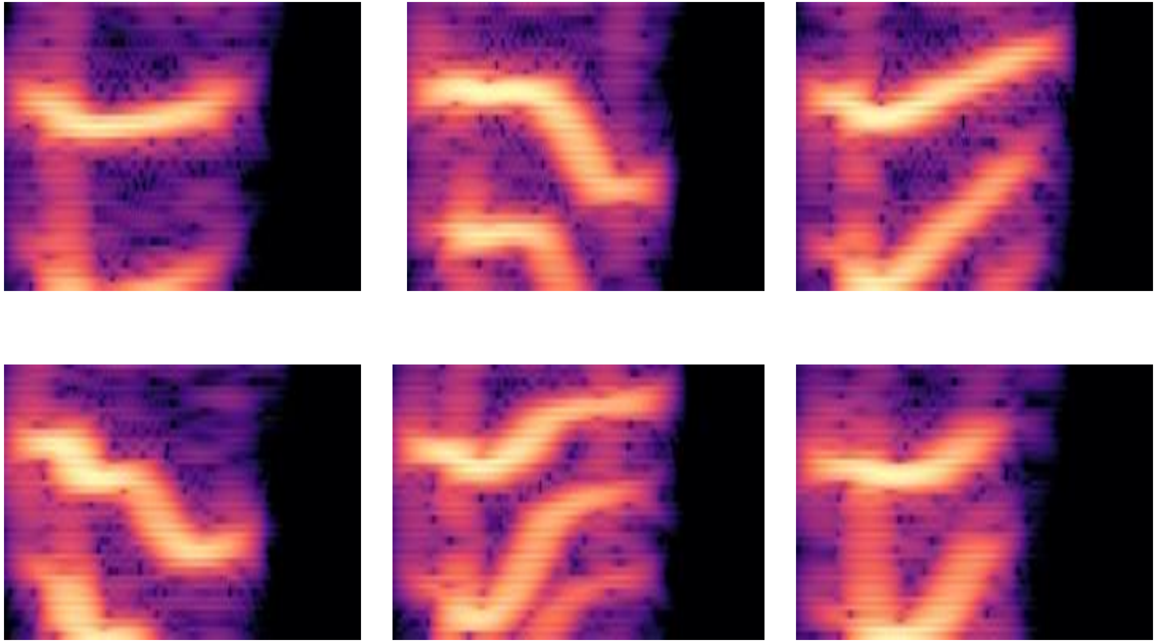


Cloud Text-To-Speech



Real Voice

ba - bá - bà - bã - bả - bạ



Building the datasets: Real Voices

PRONUNCIATION DRILLS

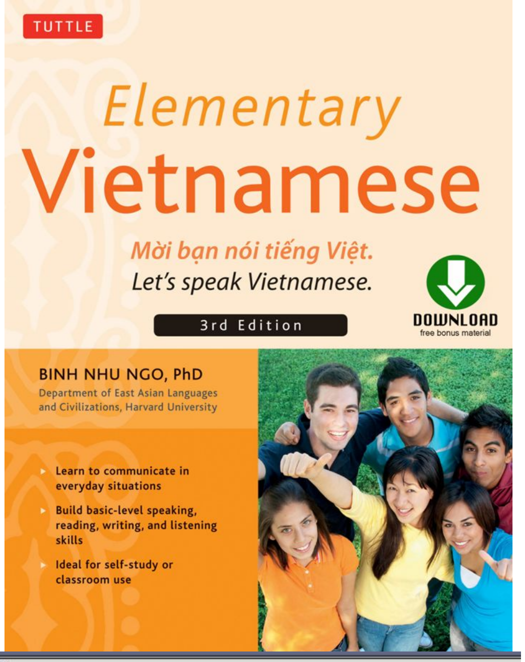
1. Listen to and repeat after the speaker. Pay attention to the production of syllables with the finals **m** and **ng** following the rounded nuclear vowels.

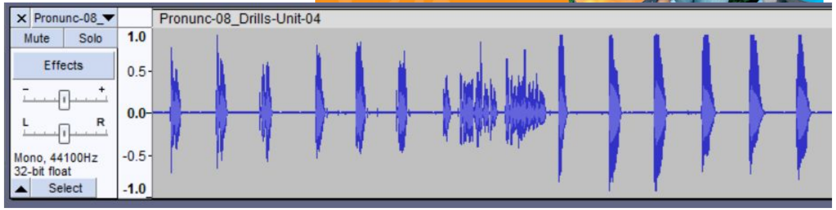
1) **um ung**	2) **ôm ông**	3) **om ong**	4) **úp úc**
đùm đùng	**xổm xổng**	**còm còng**	**cụp cục**
túm túng	**đốm đống**	**ngóm ngóng**	**đúp đúc**
bủm bủng	**chổm chổng**	**nhỏm nhỏng**	**sụp sục**
lũm lũng	**ngỗm ngỗng**	**chõm chõng**	**húp húc**
cụm cụng	**nhộm nhộng**	**khọm khọng**	**ngụp ngục**

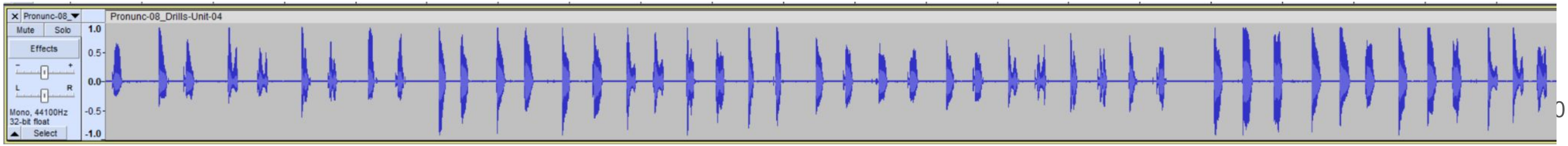
5) **ộp ộc**	6) **óp óc**
phốp phốc	**họp học**
hộp hộc	**ngóp ngóc**
lốp lốc	**cọp cọc**
độp độc	**tóp tóc**
tốp tốc	**đọp đọc**

2. Listen to and repeat after the speaker.

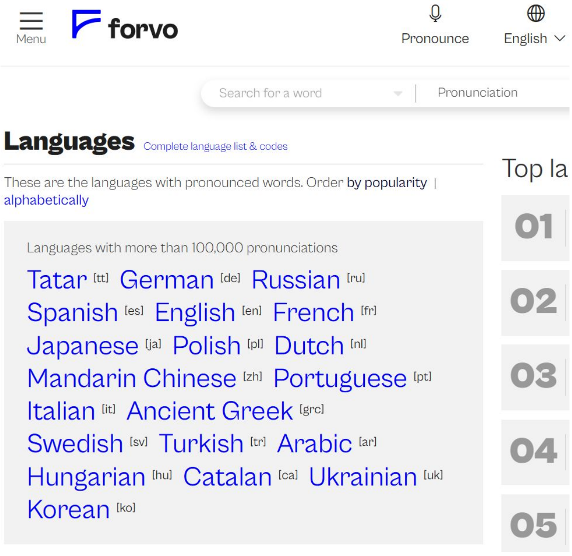
1) **ung bung dung khung cung hung lung xung sung phung trung tung mung nung rung nhung chung**
2) **mống cống tống phống bống chống nống sống đống rống hống vống lống ngống**
3) **mỏng tỏng đỏng nhỏng lỏng phỏng hỏng ngỏng dỏng bỏng**
4) **úc túc đúc múc rúc xúc lúc phúc nhúc súc cúc húc núc thúc**

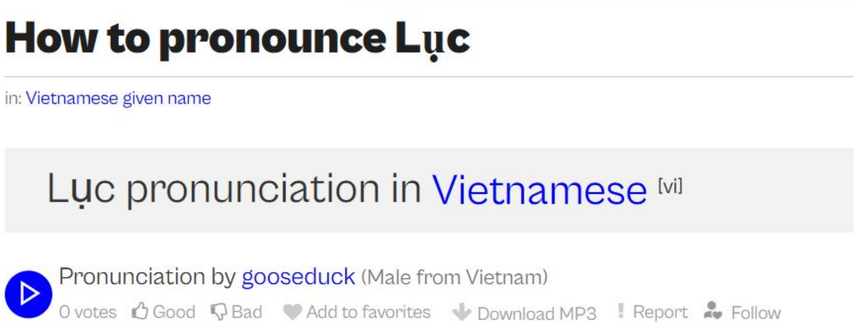






Building the datasets: Forvo Syllables





Building the datasets: Final Data

	Samples
Synthetic Data Small	1800
Synthetic Data Large	11217
Pronunciation Drills	1557
Forvo Syllables	52

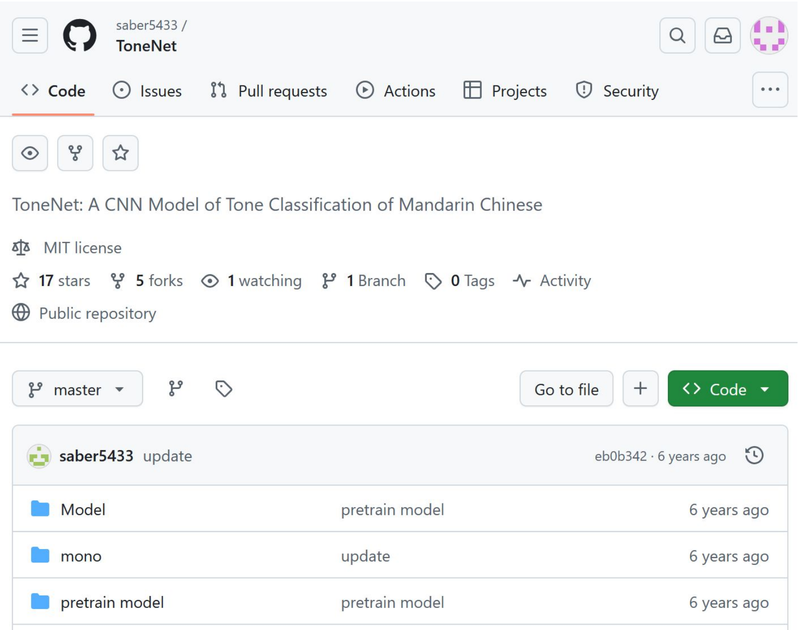
	Samples
Combined Small	3357
Combined Large	12774

Methodology: Models and Optimization

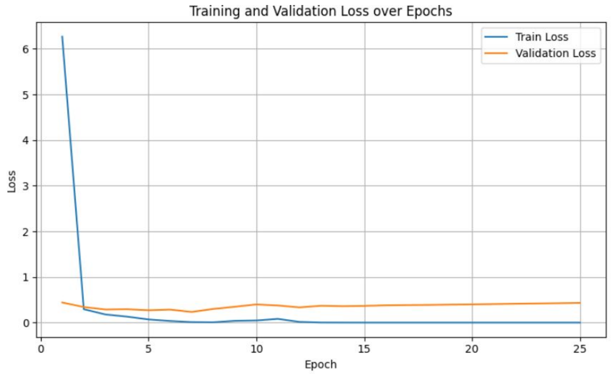
Custom CNN

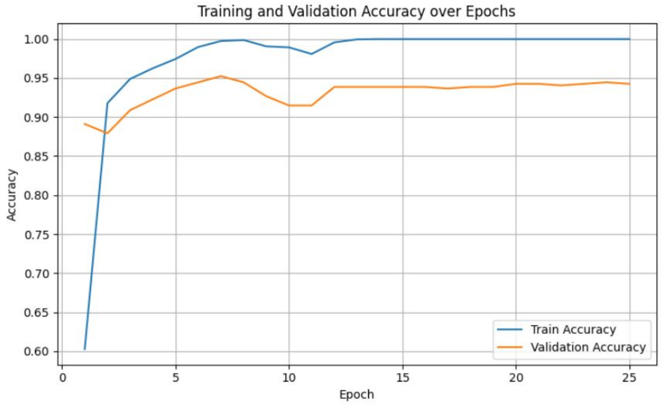
```
class SimpleToneCNN(nn.Module):
    def __init__(self, n_cls):
        super().__init__()
        self.net = nn.Sequential(
            nn.Conv2d(1, 16, 5, padding=2), nn.ReLU(), nn.MaxPool2d(2),
            nn.Conv2d(16, 32, 3, padding=1), nn.ReLU(), nn.MaxPool2d(2),
            nn.Flatten(),
            nn.Linear(32*16*56, 128), nn.ReLU(),
            nn.Linear(128, n_cls)
        )
    def forward(self, x): return self.net(x)
```

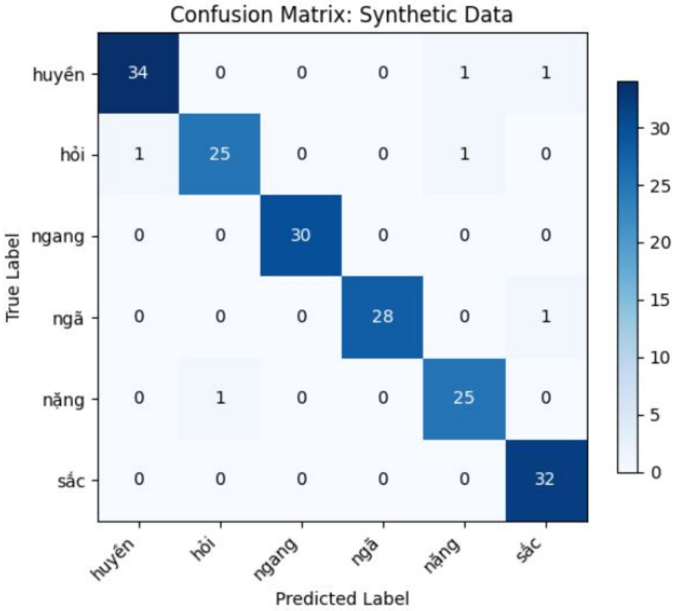
Transfer Learning: Loading Weights from ToneNet



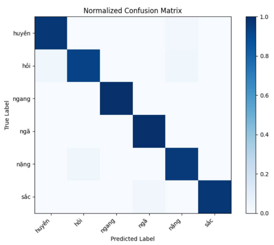
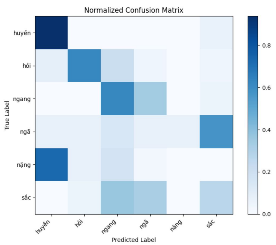
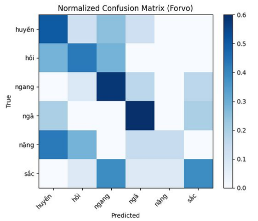
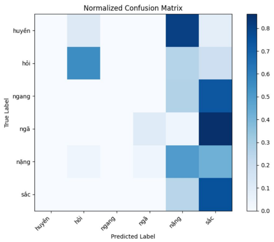
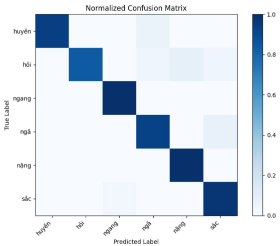
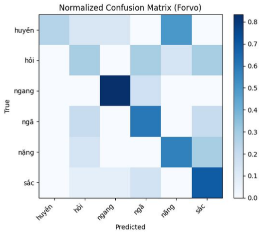
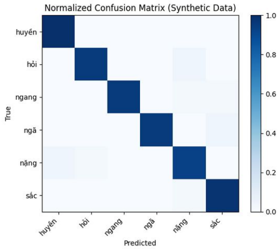
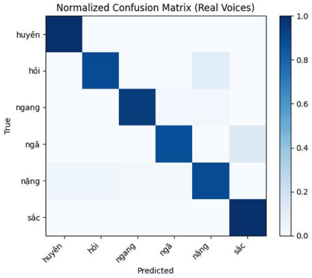
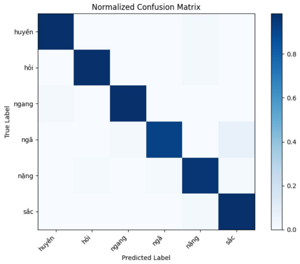
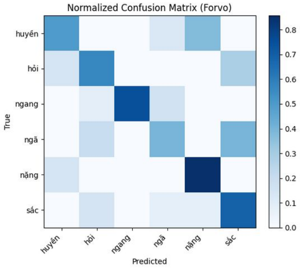
Results: Custom CNN







Results: Custom CNN

<table>
<tr><td></td><td colspan="4">Test Set</td></tr>
<tr><td>Training Set</td><td>Synthetic Data</td><td>Real Voices</td><td>Mixed</td><td>Unseen Voices</td></tr>
<tr><td>Synthetic Data</td><td>
</td><td>
</td><td></td><td>
</td></tr>
<tr><td>Real Voices</td><td>
</td><td>
</td><td></td><td>
</td></tr>
<tr><td>Mixed</td><td>
</td><td>
</td><td>
</td><td>
</td></tr>
</table>

Results: Fine-Tuning - Wav2Vec2

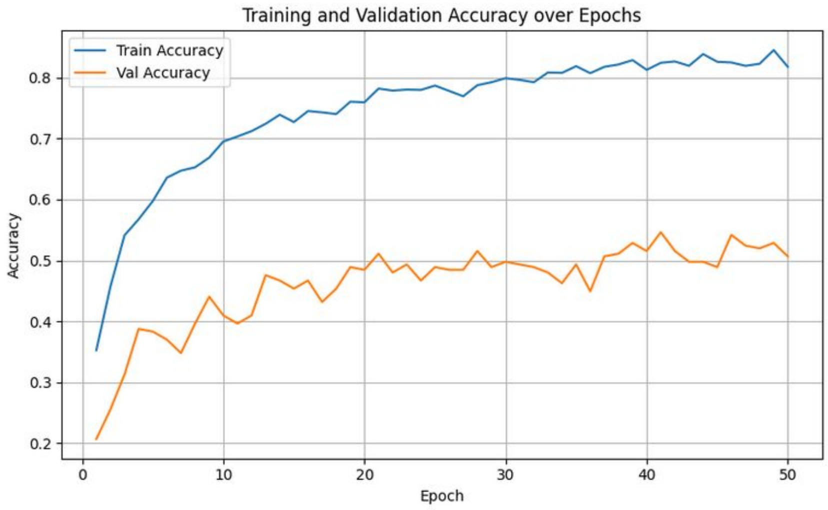
Wav2Vec2

Hugging Face

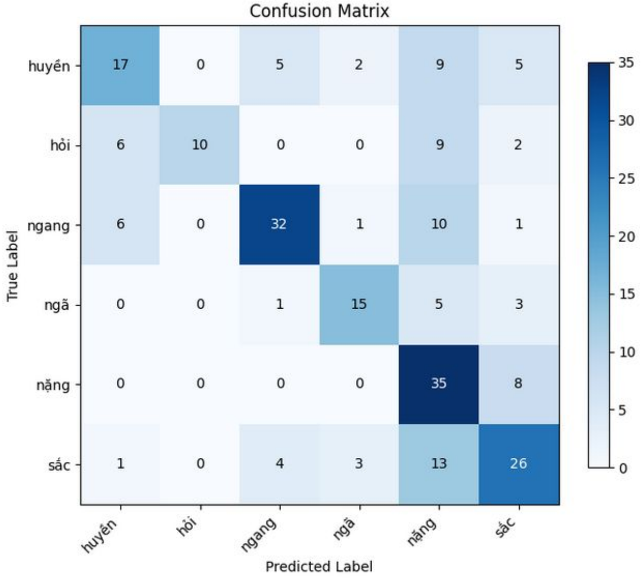
PyTorch TensorFlow Flax FlashAttention SDPA

Overview

The Wav2Vec2 model was proposed in [wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations]() by Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli.



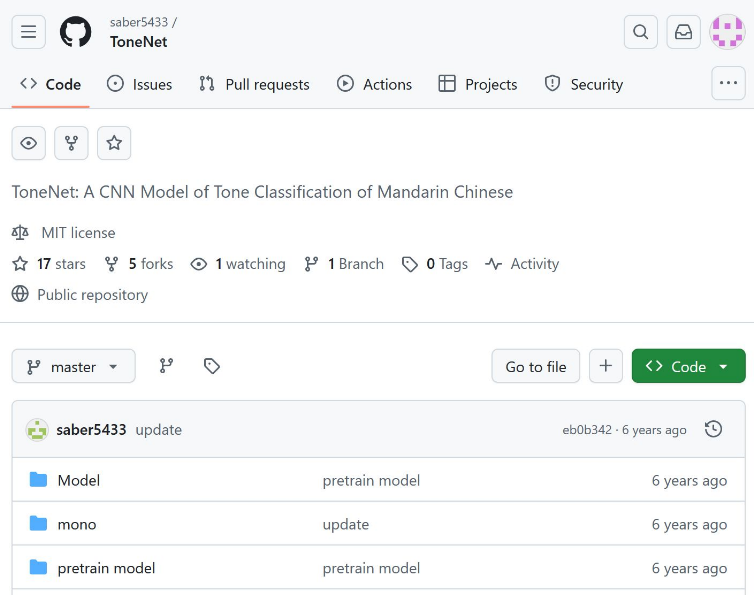
accuracy			0.59	229
macro avg	0.68	0.57	0.59	229
weighted avg	0.65	0.59	0.59	229



Name ↑

- 0000_ma.png
- 0000_ma.wav
- 0001_mi.png
- 0001_mi.wav
- 0002_ba.png
- 0002_ba.wav
- 0003_bi.png
- 0003_bi.wav
- 0004_va.png
- 0004_va.wav
- 0005_vi.png
- 0005_vi.wav

Results: Fine-Tuning - ToneNet

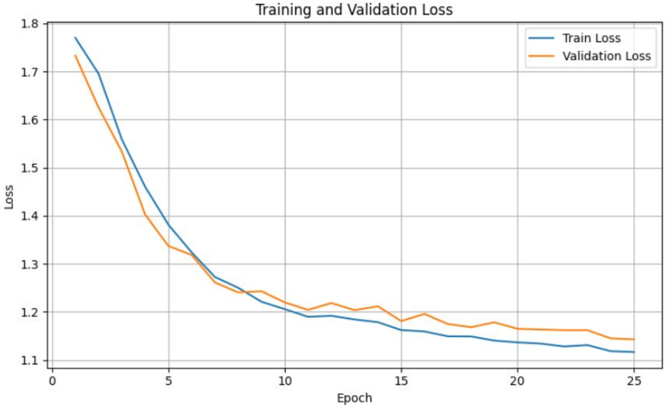


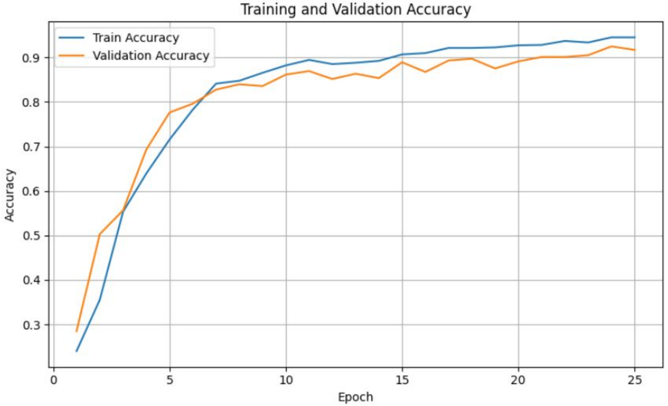
Keras → PyTorch

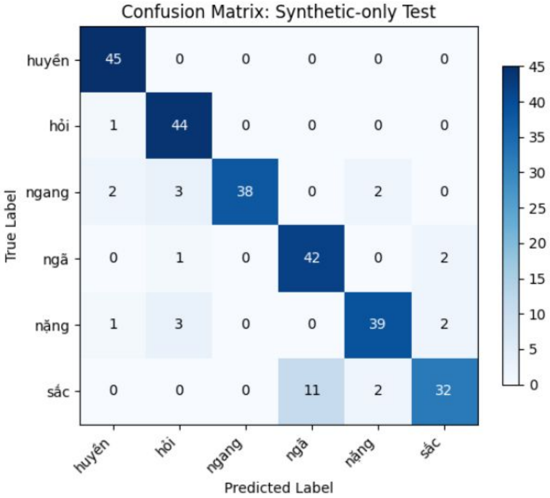
Table 1: *The ToneNet architecture, the f is the size of convolution kernels and the s is stride.*

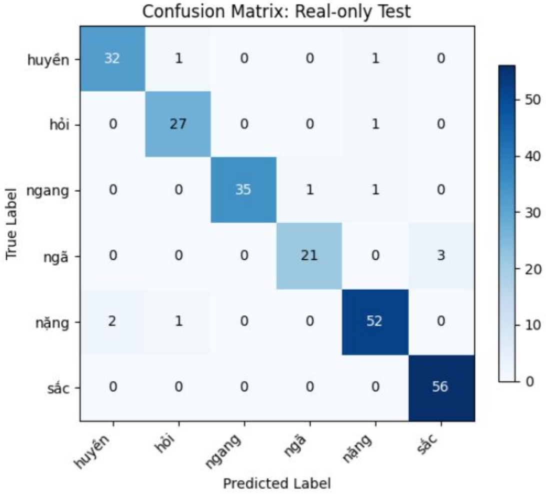
Name	**ToneNet**
Input	Image
Part-1	Conv2d(f=5 × 5 × 64, s=3) BatchNormalization MaxPooling2d(f=3 × 3, s=3)
Part-2	Conv2d(f=3 × 3× 128, s=1) BatchNormalization MaxPooling2d(f=2 × 2, s=2) Conv2d(f=3 × 3 × 256, s=1) BatchNormalization MaxPooling2d(f=2 × 2, s=2) Conv2d(f=3 × 3 × 256, s=1) BatchNormalization MaxPooling2d(f=2 × 2, s=2) Conv2d(f=3 × 3 × 512, s=1) BatchNormalization MaxPooling2d(f=2 × 2, s=2)
Flatten	Flatten
Part-3	FC-1024 BatchNormalization FC-1024 BatchNormalization FC-4 SoftMax

Results: Fine-Tuning - ToneNet



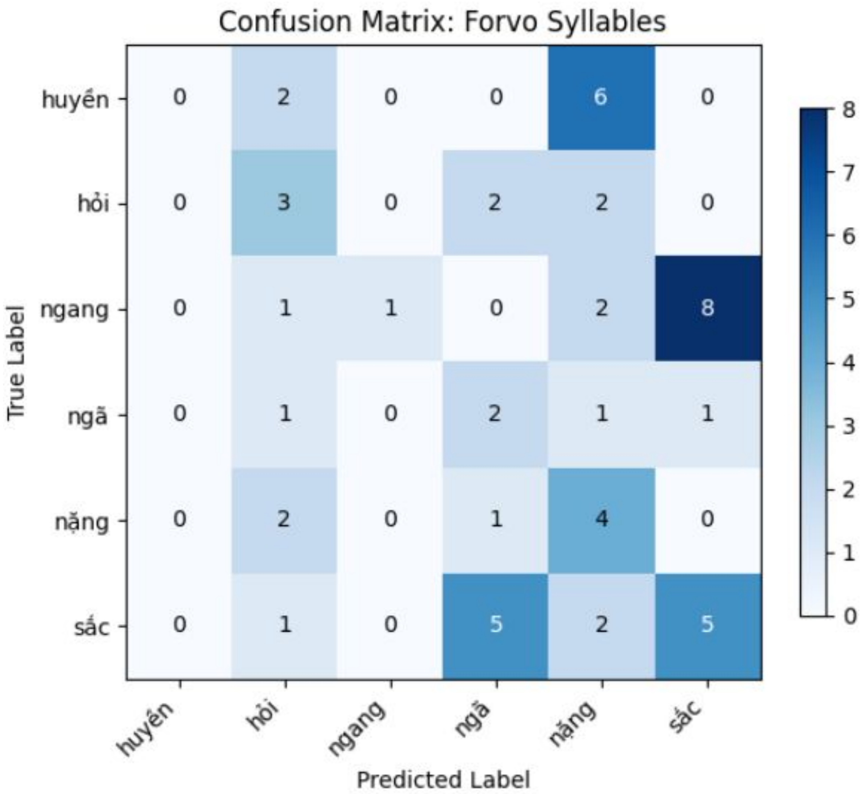






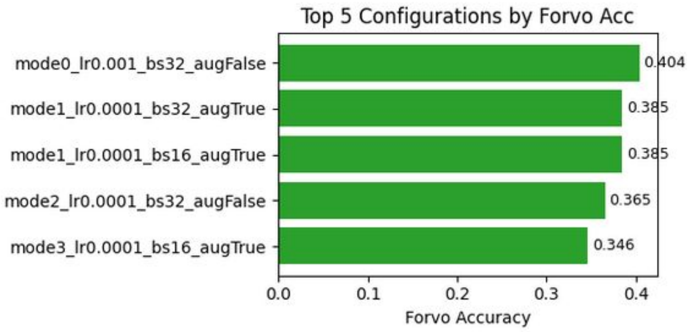
Results: Fine-Tuning - ToneNet

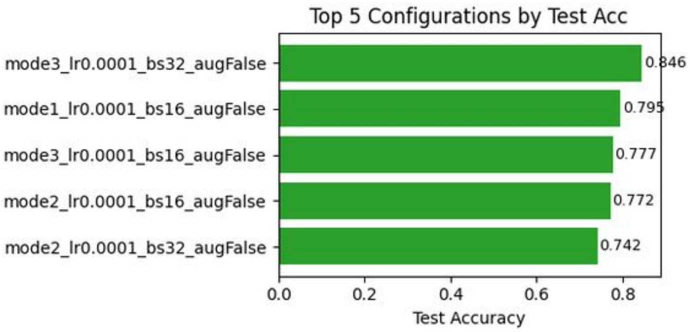
0.29 accuracy

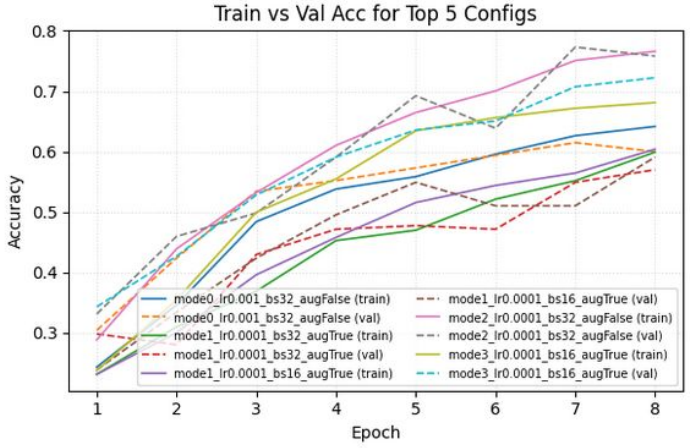


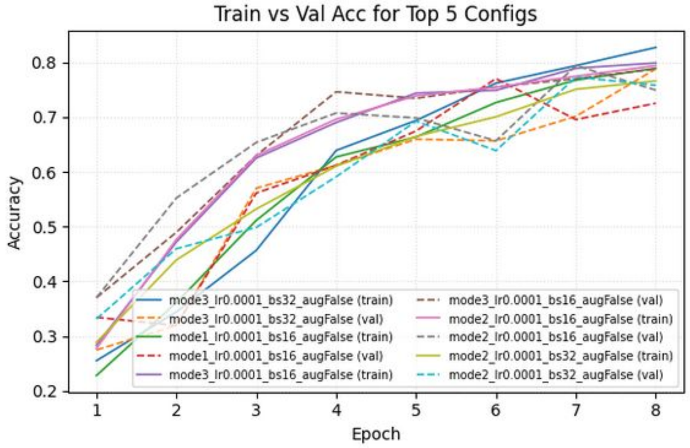
0.29 accuracy

Results: ToneNet and Performance on Unseen Voices

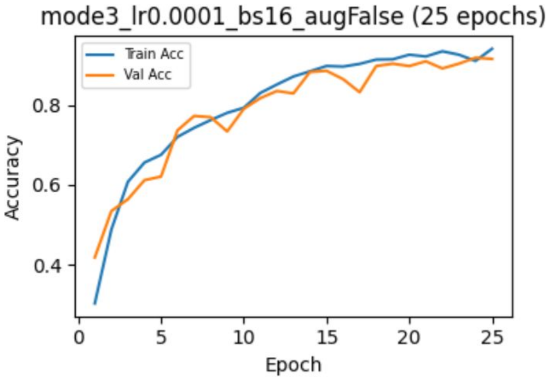


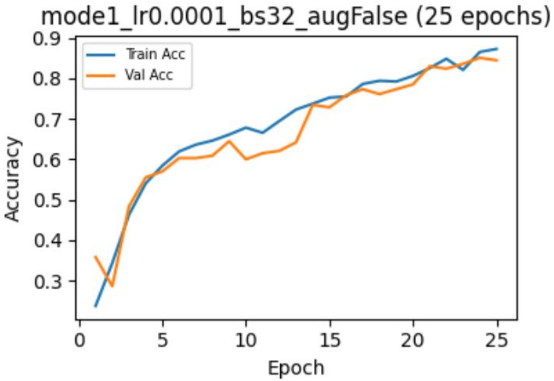


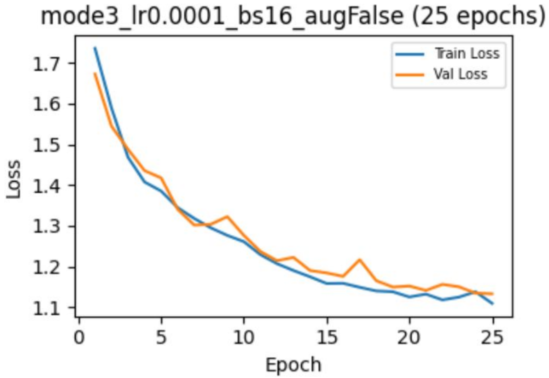


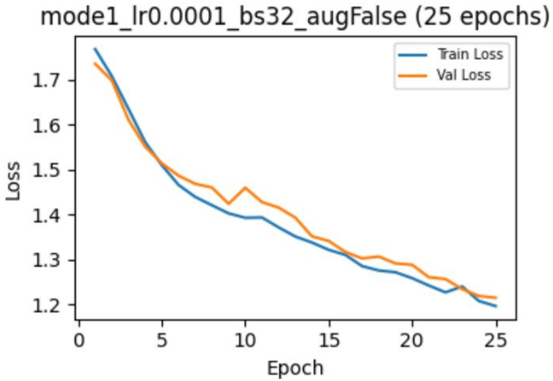


Results: ToneNet and Performance on Unseen Voices









Another approach: Data Augmentation (Park, 2019)

IDEA: Randomly masks out contiguous frequency and time regions in a mel-spectrogram to simulate missing information

Frequency Masking

- Chooses ≈ (freq_mask_pct × 64) consecutive mel bins
- Sets those rows to –80 dB (silence) across all time frames

Time Masking

- Chooses ≈ (time_mask_pct × 225) consecutive time frames
- Sets those columns to –80 dB (silence) across all frequency bins

```
def spec_augment(spec: np.ndarray,
                 time_mask_pct: float = 0.10,
                 freq_mask_pct: float = 0.10) -> np.ndarray:
    H, W = spec.shape
    # Frequency mask
    n_freq = int(H * freq_mask_pct)
    if n_freq > 0:
        f0 = random.randint(0, H - n_freq)
        spec[f0 : f0 + n_freq, :] = -80.0

    # Time mask
    n_time = int(W * time_mask_pct)
    if n_time > 0:
        t0 = random.randint(0, W - n_time)
        spec[:, t0 : t0 + n_time] = -80.0

    return spec
```

Results: Data Augmentation

	Testing Set: Forvo Syllables	
Training Set	Without Data Augmentation	With Data Augmentation
Synthetic Data	0.4423	0.3846
Real Voices	0.5769	0.6154
Mixed	0.5192 (large set) 0.6538 (small set)	0.6923

Final Reflections and Conclusions

1. Syllables (and specially tones) in isolation behave way differently than in contact, so data from full sentences corpus poses a difficulty for the task.
2. Existing models like Wav2Vec do not focus on tone recognition and transfer learning has been proven challenging.
3. A combination of synthetic+real data has been seen to be effective and perform well on new data.
4. Sometimes the challenge can be in the data more than in the Deep Learning architecture and/or hyperparameter tuning.

Future Steps

1. Gather more data from real speakers: variety is important.
2. Explore more data augmentation techniques.
3. Extend the work to other tonal languages.
4. Keep exploring transfer learning and fine-tuning.
5. Deploy the model into an application.

Audio Course
Search documentation Ctrl+K
EN 418

UNIT 0. WELCOME TO THE COURSE!
UNIT 1. WORKING WITH AUDIO DATA
UNIT 2. A GENTLE INTRODUCTION TO AUDIO APPLICATIONS
UNIT 3. TRANSFORMER ARCHITECTURES FOR AUDIO
UNIT 4. BUILD A MUSIC GENRE CLASSIFIER
UNIT 5. AUTOMATIC SPEECH RECOGNITION
UNIT 6. FROM TEXT TO SPEECH
UNIT 7. PUTTING IT ALL TOGETHER
UNIT 8. FINISH LINE
COURSE EVENTS

Introduction to audio data

By nature, a sound wave is a continuous signal, meaning it contains an infinite number of signal values in a given time. This poses problems for digital devices which expect finite arrays. To be processed, stored, and transmitted by digital devices, the continuous sound wave needs to be converted into a series of discrete values, known as a digital representation.

References

Gao, Q., Sun, S., & Yang, Y. (2019). ToneNet: A CNN model of tone classification of Mandarin Chinese. In *Proceedings of Interspeech 2019* (pp. 3367–3371).

Bui, T. H. (2020). Vietnamese voice classification based on deep learning approach. *International Journal of Machine Learning and Networked Collaborative Engineering, 4*(4), 171–180.

Huang, H., Hu, Y., & Xu, H. (2017). Mandarin tone modeling using recurrent neural networks [Preprint]. *arXiv*. arXiv:1711.01946

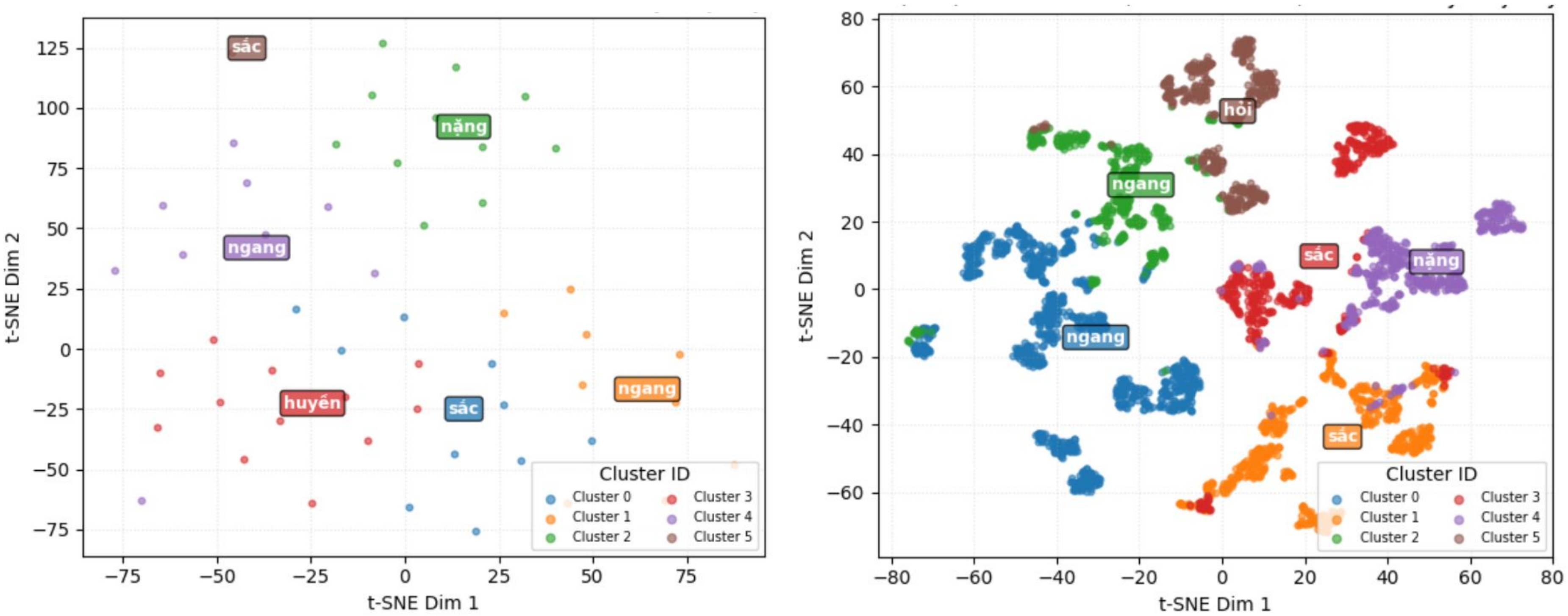
Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. In *Proceedings of Interspeech 2019* (pp. 2613–2617).

Tang, J., & Li, M. (2021). End-to-end Mandarin tone classification with short-term context information [Preprint]. *arXiv*. arXiv:2104.05657

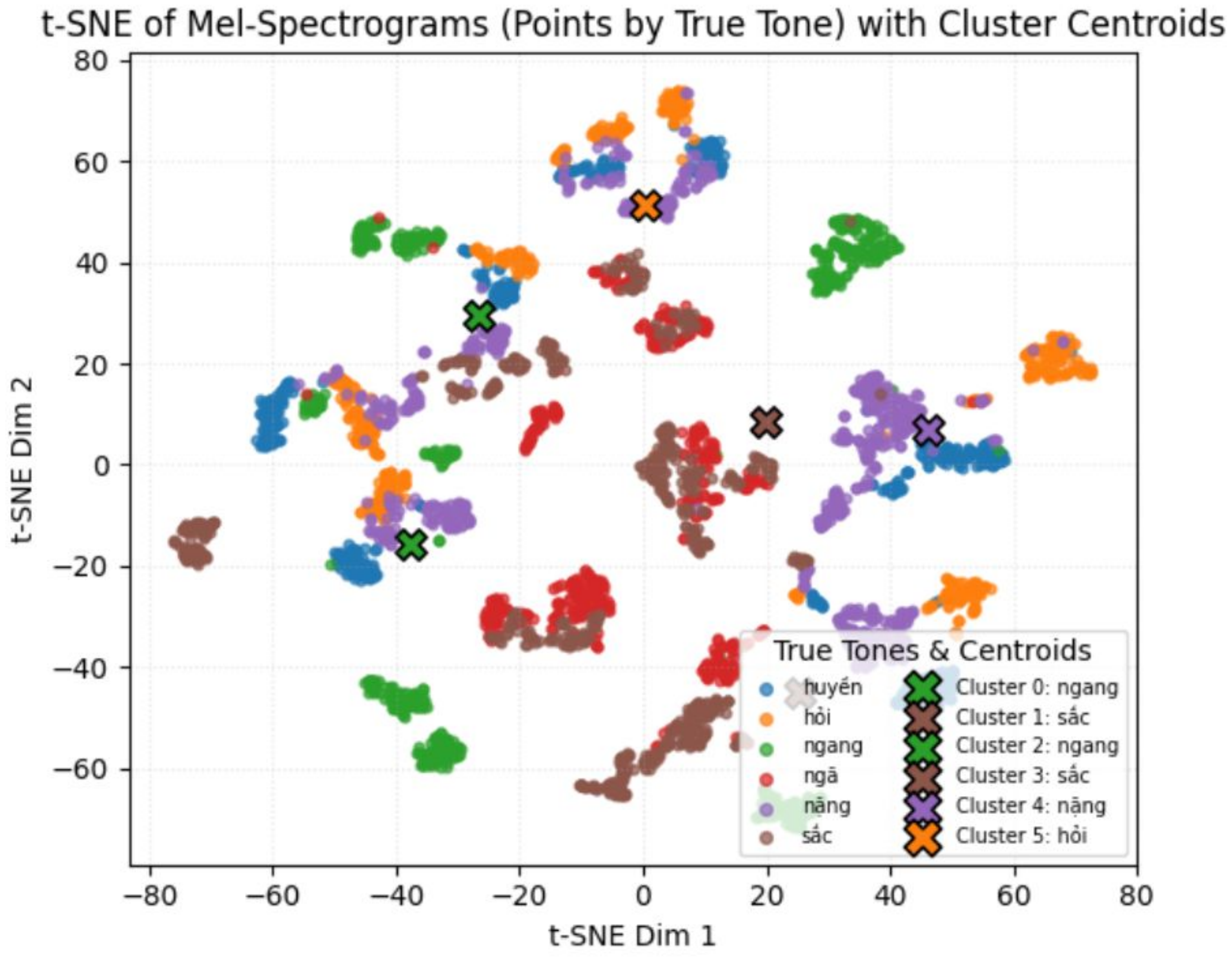
```
SimpleToneCNN(
  (net): Sequential(
    (0): Conv2d(1, 16, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Flatten(start_dim=1, end_dim=-1)
    (7): Linear(in_features=28672, out_features=128, bias=True)
    (8): ReLU()
    (9): Linear(in_features=128, out_features=6, bias=True)
  )
)
```

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [1, 16, 64, 224]             416
              ReLU-2          [1, 16, 64, 224]               0
         MaxPool2d-3          [1, 16, 32, 112]               0
            Conv2d-4          [1, 32, 32, 112]           4,640
              ReLU-5          [1, 32, 32, 112]               0
         MaxPool2d-6           [1, 32, 16, 56]               0
           Flatten-7                [1, 28672]               0
            Linear-8                  [1, 128]       3,670,144
              ReLU-9                  [1, 128]               0
           Linear-10                    [1, 6]             774
================================================================
```

```
================================================================
Total params: 3,675,974
Trainable params: 3,675,974
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.05
Forward/backward pass size (MB): 6.13
Params size (MB): 14.02
Estimated Total Size (MB): 20.20
----------------------------------------------------------------
Total parameters: 3,675,974
```



sắc
nặng
ngang
huyền
sắc
ngang
Cluster ID
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
t-SNE Dim 1
t-SNE Dim 2
hỏi
ngang
sắc
nặng
ngang
sắc


t-SNE of Mel-Spectrograms (Points by True Tone) with Cluster Centroids
t-SNE Dim 2
t-SNE Dim 1
80
60
40
20
0
−20
−40
−60
−80
−60
−40
−20
0
20
40
60
80
True Tones & Centroids
huyền
hỏi
ngang
ngã
nặng
sắc
Cluster 0: ngang
Cluster 1: sắc
Cluster 2: ngang
Cluster 3: sắc
Cluster 4: nặng
Cluster 5: hỏi