Deep Learning for Voice Faker

PROJECT PRESENTATION

Group 22：張又仁 陳志誠 江詠筑

TABLE OF CONTENTS

01

INTRODUCTION

An introduction of our project and why we think it is interesting.

PROJECT MOTIVATION and GOAL

Motivation 1

Deepface is popular
we can combine them together

Motivation 2

Help people with visual
or reading impairments

Goal 1

construct simple
identification model

Goal 2

help the lazy guys and the
speaking handicapped
to have a better life

02

ABOUT THE PROJECT

Project Result and the Colab Implementation

PROJECT STAGES

	Classification	Speech-To-Text
Dataset	Urbansound8k	Libasound train-100
Model	Audio to MelSpectrom & eliminate noise	Greedy decoding & CTCloss + deep speech 3
Accuracy	70%	70%
Benchmark	70~79%	80~84%

Colab Result

AI22.ipynb

檔案 編輯 檢視畫面 插入 執行階段 工具 說明 系統不會儲存變更

+ 程式碼 + 文字 複製到雲端硬碟

RAM 磁碟 編輯 共用

[12]

```python
            if (new_channel == 1):
                # Convert from stereo to mono by selecting only the first channel
                resig = sig[:1, :]
            else:
                # Convert from mono to stereo by duplicating the first channel
                resig = torch.cat([sig, sig])

            return ((resig, sr))
def spectro_gram(aud, n_mels=64, n_fft=1024, hop_len=None,top=80):
            sig, sr = aud
            top_db = top

        # spec has shape [channel, n_mels, time], where channel is mono, stereo etc
            spec = transforms.MelSpectrogram(
            sr, n_fft=n_fft, hop_length=hop_len, n_mels=n_mels)(sig)

        # Convert to decibels
            spec = transforms.AmplitudeToDB(top_db=top_db)(spec)
            return (spec)
for response in os.walk(file_path):
    for x in response[2]:
        if x[-3:]=='txt':
            continue
        s,rate=rechannel(torchaudio.load(file_path+'/'+x),1)
        spectrograms = []
        w = valid_audio_transforms(s).squeeze(0).transpose(0, 1)
        spectrograms.append(w)
        s = nn.utils.rnn.pad_sequence(spectrograms, batch_first=True).unsqueeze(1).transpose(2, 3)
        device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
        s=torch.squeeze(s,2)
        input=s.to(device)
        output = model(input)
        output = F.log_softmax(output, dim=2)
        output = output.transpose(0, 1) # (time, batch, n_class)
        # Get the predicted class with the highest score
        decoded_preds, decoded_targets = GreedyDecoder(output.transpose(0, 1), [], 1,out=True)
        print(file_path+'/'+x,decoded_preds)
```

```
test/0.flac ['it was nowmided joly and the plag which had cheefly raged at the other end o the town and as i said before in the parisheus of sant chiles saint and drus  hol firn']
```

Test result

Load pretrained model

[8]

```python
import gdown
path="https://drive.google.com/u/1/uc?id=1043wWFXJm5XzFL4wa7wf8nmD2qKZScfw&export=download"
output ='/content/sounds.pth'
gdown.download(path, output,quiet=True)
model.load_state_dict(torch.load(output,map_location=torch.device('cpu')))
```

```
<All keys matched successfully>
```

[9]

```python
path="https://drive.google.com/u/1/uc?id=1dBGrwgOKAQOFi6VVtpAnzLat7CHbB3nb&export=download"
output ='test.7z'
gdown.download(path, output,quiet=True)
```

```
'test.7z'
```

[10]

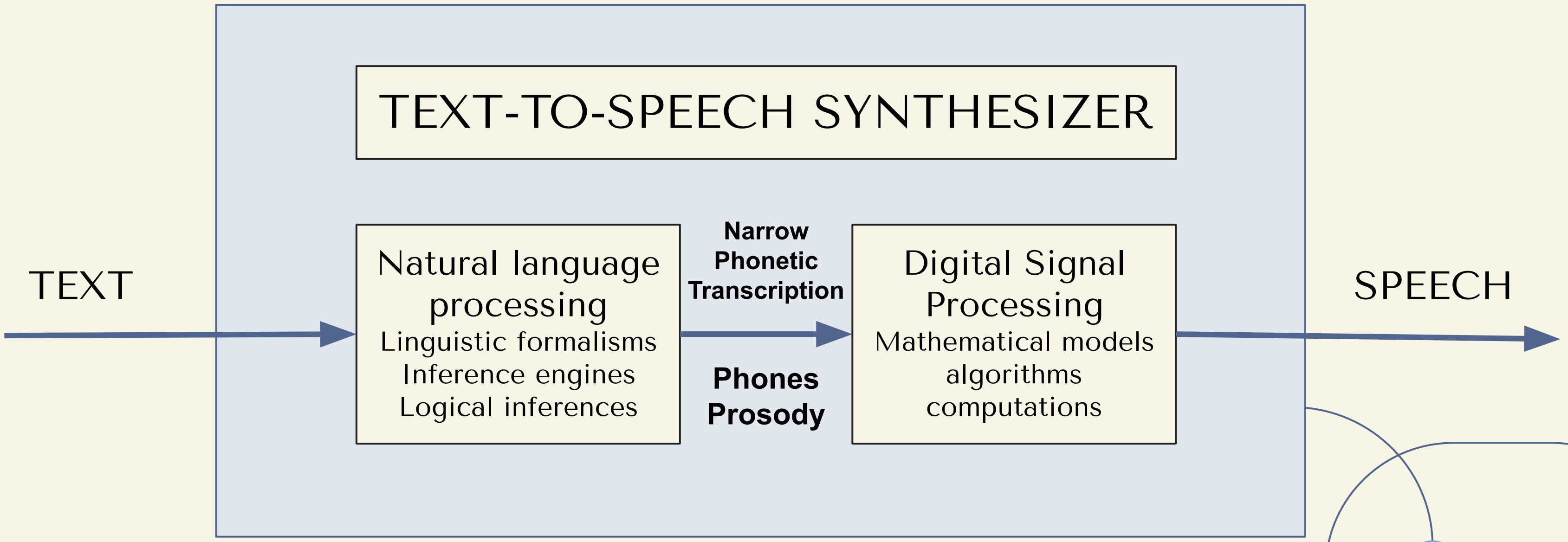
```
!p7zip -d 'test.7z'
```

03

Text-To-Speech

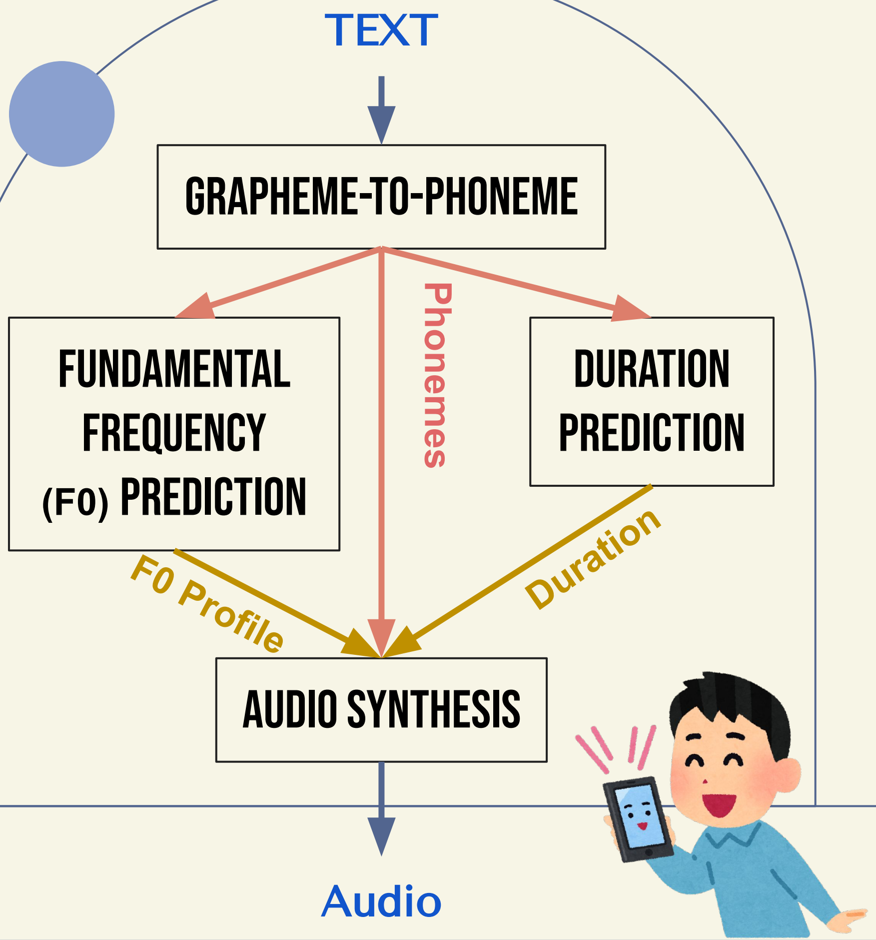
Text-to-speech (TTS) lets your devices read text to you. Text-to-speech is used not only to **help people with visual or reading impairments**, but also **convenient for those who are lazy to talk** .

Speech Synthesis (TTS) Technology

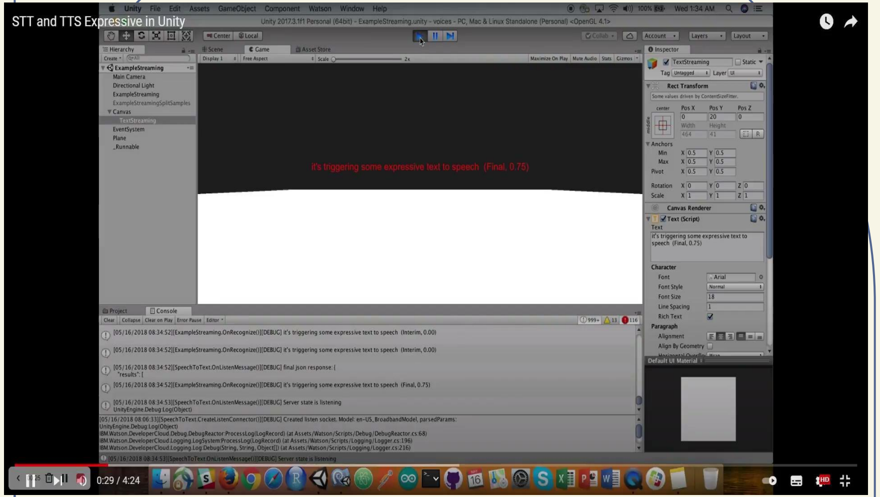


How to implement ?

1. Deep voice 3 open source code and well-trained model for single and multiple speaker TTS https://github.com/r9y9/deepvoice3_pytorch can be run on colab
2. Deep Voice: Real-time Neural Text-to-Speech https://proceedings.mlr.press/v70/arik17a/arik17a.pdf



From STT to TTS



STT and TTS Demo in IBM Watson SDK for Unity | Github Source :
https://github.com/rustyoldrake/ibm_watson_unity/blob/master/ExampleStreaming_plus_Text_to_Speech_expressive.cs

After STT completes the process, there are at least **four further processes** to be carried out.

1. **ASR (speech recognition)**
2. **NLU (natural language understanding)**
3. **DST (dialogue state control)**
4. **NLG (dialogue generation)**

and eventually arrive at **TTS (Text To Speech)**

04

ABOUT

Future Work

Outlook of Other techniques

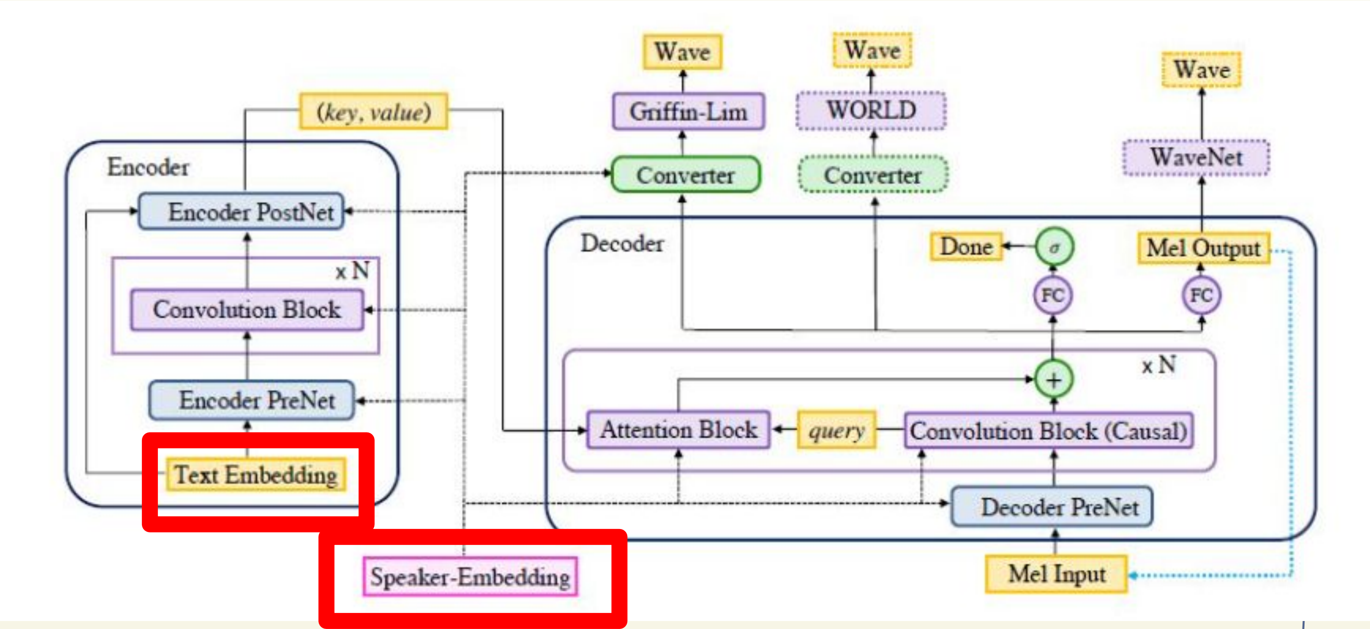
Voice cloning technology

Research on a practical technique for training TTS (Text to Speech) model with **few samples**

How to do?

pretrained + fine-tone

Multi-speaker model—Deep voice 3



cloning

Speaker adaptation

fine-tune a trained multi-speaker model for an unseen speaker using a few audio-text pairs

Speaker encoding

Train a separate network to generate new speaker embeddings, which are then used for multi-speaker generation models

conclusion

For the TTS task, the generation network of a single speaker requires nearly 20 hours of training data, but when voice cloning an unseen new person speaks, it only takes a few minutes or a few seconds.

Reference

- Neural Voice Cloning with a Few Samples / Jitong Chen,Kainan Peng,and Wei Ping
  In NIPS 2018 https://arxiv.org/pdf/1802.06006.pdf
- Deep4SNet: deep learning for fake speech classification / Dora M.,Yohanna,Diego,Gonzalo
  https://www.sciencedirect.com/science/article/pii/S0957417421008770
- Audio Deep Learning Made Simple: Sound Classification, Step-by-Step | by Ketan Doshi
- Audio Signal Processing and Recognition
- AFE characteristics proposed by the European Telecommunications Standards Institute
  https://www.etsi.org/deliver/etsi_es/201100_201199/201108/01.01.03_60/es_201108v010103p.pdf
- GitHub - mozilla/DeepSpeech
- The state-of-art benchmark accuracy of our project's datasets
  - Urbansound8k (Classification) / 79%
  - UrbanSound8K Benchmark (Environmental Sound)Classification | Papers With Code
  - Libasound train-100 (Audio-to-text) / 84%
    Global speech-to-text transcript error rating 2020 | Statista

THANKS

This is the end of our presentation

By Group 22 張又仁 陳志誠 江詠筑