Krzysztof Jeschke 171566

Diagnostyka systemów

Aplikacja na system Windows umożliwiająca translację wpisanego tekstu na mowę

1. Opis projektu

Projekt umożliwia wykonanie translacji wpisanego tekstu na ludzką mowę w języku angielskim. Wykonana aplikacja posiada interfejs użytkownika, który pozwala na wprowadzenie tekstu oraz odtworzenie go po naciśnięciu przycisku. Wykorzystane technologie:

- Python

- TensorFlowTTS z akceleracją NVIDIA Cuda 10.1

- Google Colab

Projekt pierwotnie zakładał wykonanie własnego dataset'u i wytrenowanie własnego translatora, jednak z powodu sporych problemów, głownie spowodowanych niską mocą obliczeniową dostępnego urządzenia oraz ubogim zbiorem audio, nie udało się wyuczyć modelu, który byłby w stanie dokonać takiej translacji w języku polskim. Dalszy opis sprawozdania dotyczy mimo wszystko stworzonego dataset'u, procesu uczenia oraz wyników, gdyż stanowiły one drogę autora do końcowej aplikacji.

2. Dataset

Na potrzeby modelu został stworzony zbiór 164 kilkusekundowych nagrań głosowych (nagranych przez autora), odpowiednio sformatowanych oraz plik *.csv zwierający informacje o nazwie pliku audio i odpowiadające mu zdania wypowiadane w nagraniu. Format danych został stworzony zgodnie ze standardem ljspeech:

ścieżka_pliku | Wypowiadane zdanie. | Wypowiadane zdanie z rozwiniętymi skrótami, liczbami.

Przykładowe wartości z pliku:

```
LJ001-0107|Głupstwo, mój drogi.|Głupstwo, mój drogi.
LJ001-0108|Przecież mi obiecałaś…|Przecież mi obiecałaś…
LJ001-0109|Henry zaśmiał się zuchwale.|Henry zaśmiał się zuchwale.
```

Następnie utworzony dataset został przekonwertowany na format wykorzystywany w uczeniu modelu za pomocą TensorFlowTTS wykorzystując:

- tensorflow-tts-preprocess

- tensorflow-tts-normalize

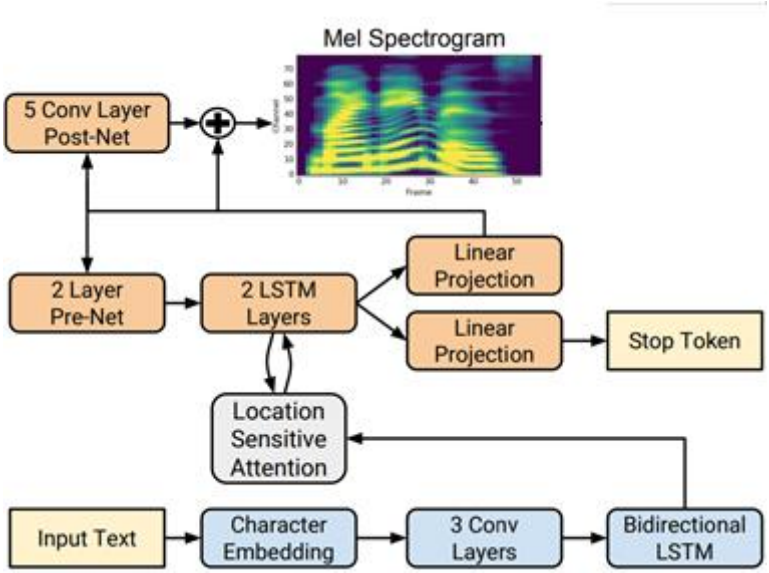
3. Uczenie

Podczas pracy nad działającym modelem zostało wykorzystanych kilka architektur translacji tekstu na mowę:

- Tacotron 2

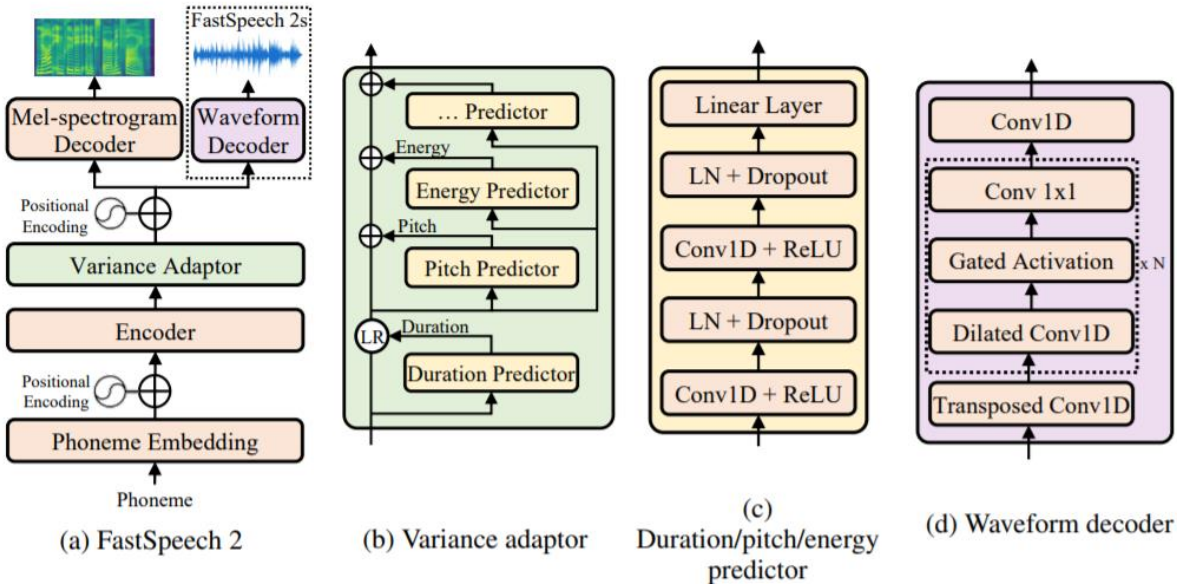
  Sam tacotron jest zdolny do przetworzenia wejścia w postaci tekstu na spektrogram Mel

(jest to sposób na wizualne przedstawienie głośności lub amplitudy sygnału na skali Mel). Po wygenerowaniu takiego spektrogramu potrzebny jest dodatkowy model zamieniający go na wyjście audio np. HiFi-GAN lub MelGAN.



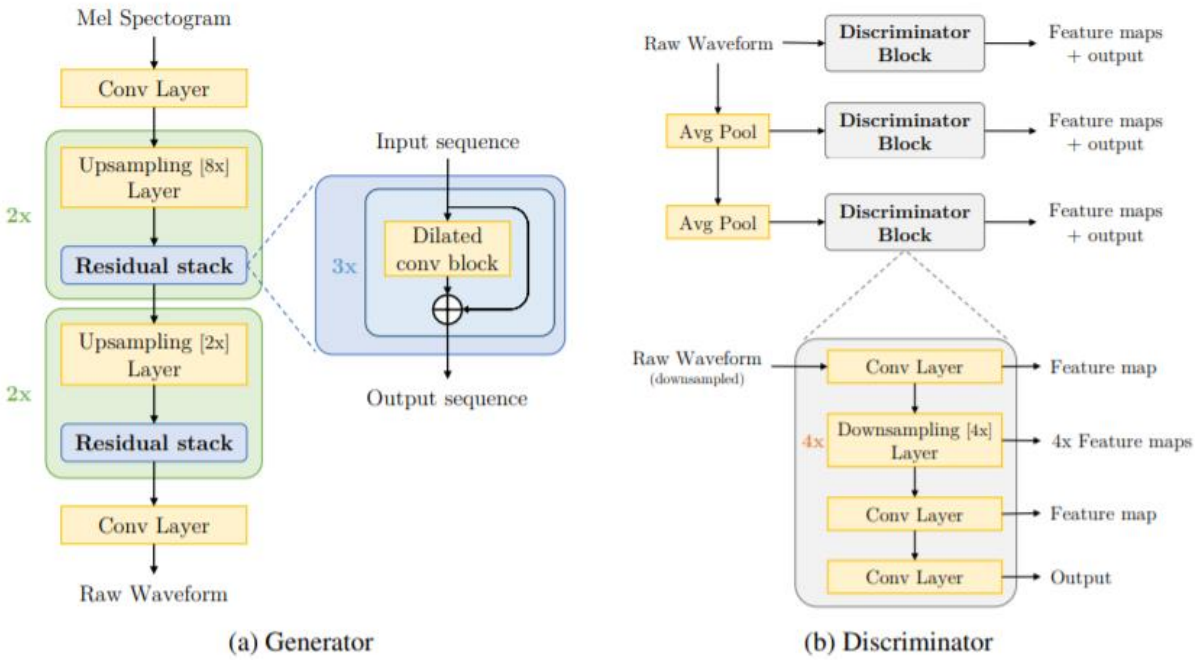
- FastSpeech i FastSpeech 2

Model FastSpeech do uczenia potrzebuje wcześniej wyuczony model zdolny do generowania spektrogramów mel (np. Tacotron 2) do procesu zwanego *"knowledge distillation"*, czyli transferu wiedzy z dużego modelu do mniejszego. FastSpeech 2 eliminuje taką konieczność, co znacznie przyspiesza uczenie modelu, jednak nadal wymaga ekstrakcji czasu trwania dźwięku wykorzystując wcześniej wyuczony model. Wyjściem takich modeli są ponownie spektrogramy mel, dlatego w celu uzyskania audio należy użyć modelu zamieniającego je na wyjście audio np. HiFi-GAN lub MelGAN.



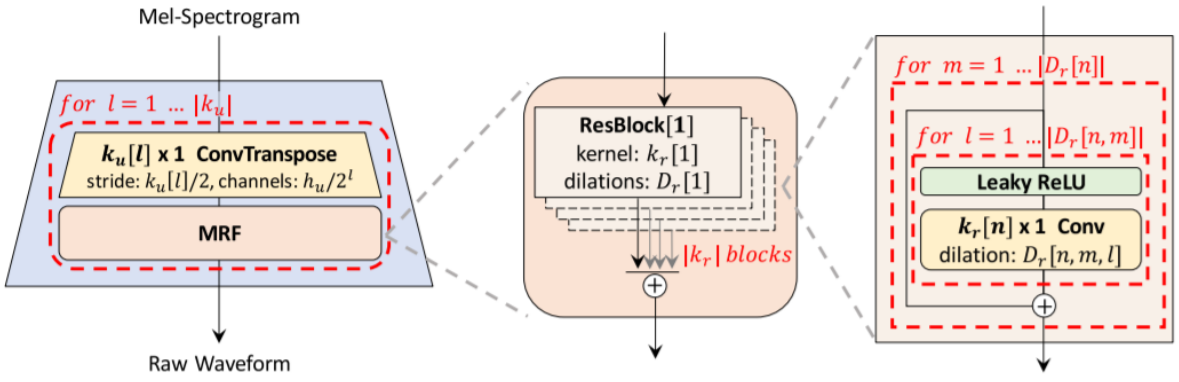
- MelGAN

Model MelGAN wykorzystywany jest na wyjściu modeli generujących spektrogramy mel. Używa się go do zamiany wspomnianych spektrogramów na dźwięk audio. Dzięki wykorzystaniu podejścia GAN (generative adversarial networks), model zarówno bardzo szybko się uczy, jak i działa.



- HiFi-GAN

Model ten, podobnie jak opisywany wcześniej MelGAN, przyjmuje na wejście spektrogram mel i konwertuje go na odpowiedni plik audio. Według jego autorów [5] model HiFi-GAN osiąga zarówno wyższą efektywność obliczeniową jak i jakość wyjściowych próbek.

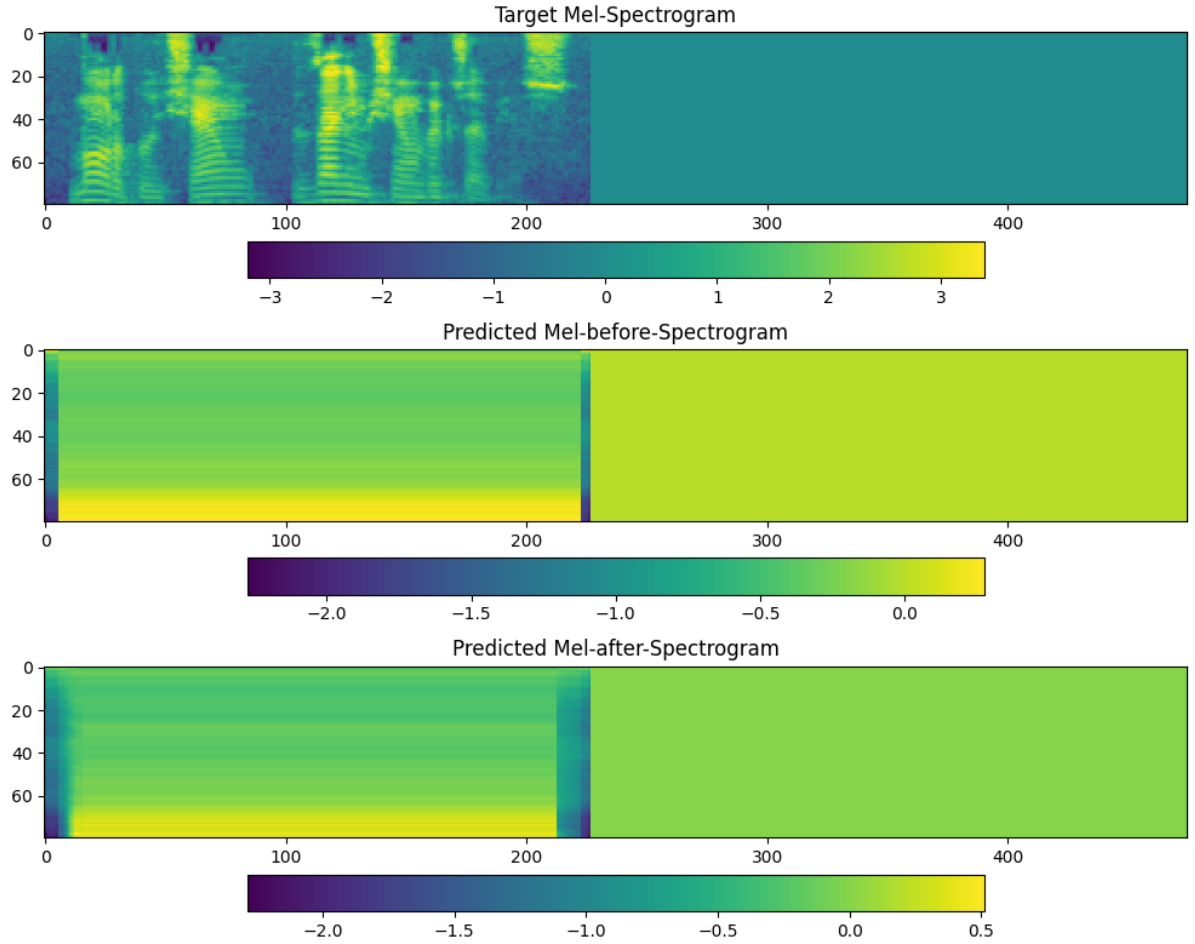


4. Wyniki

Konfiguracja środowiska w celu uzyskania kompatybilnych bibliotek i wersji oprogramowania zajęła znaczną część czasu. Autor napotkał wiele błędów w konfiguracji TensorFlowTTS oraz NVIDIA Cuda.

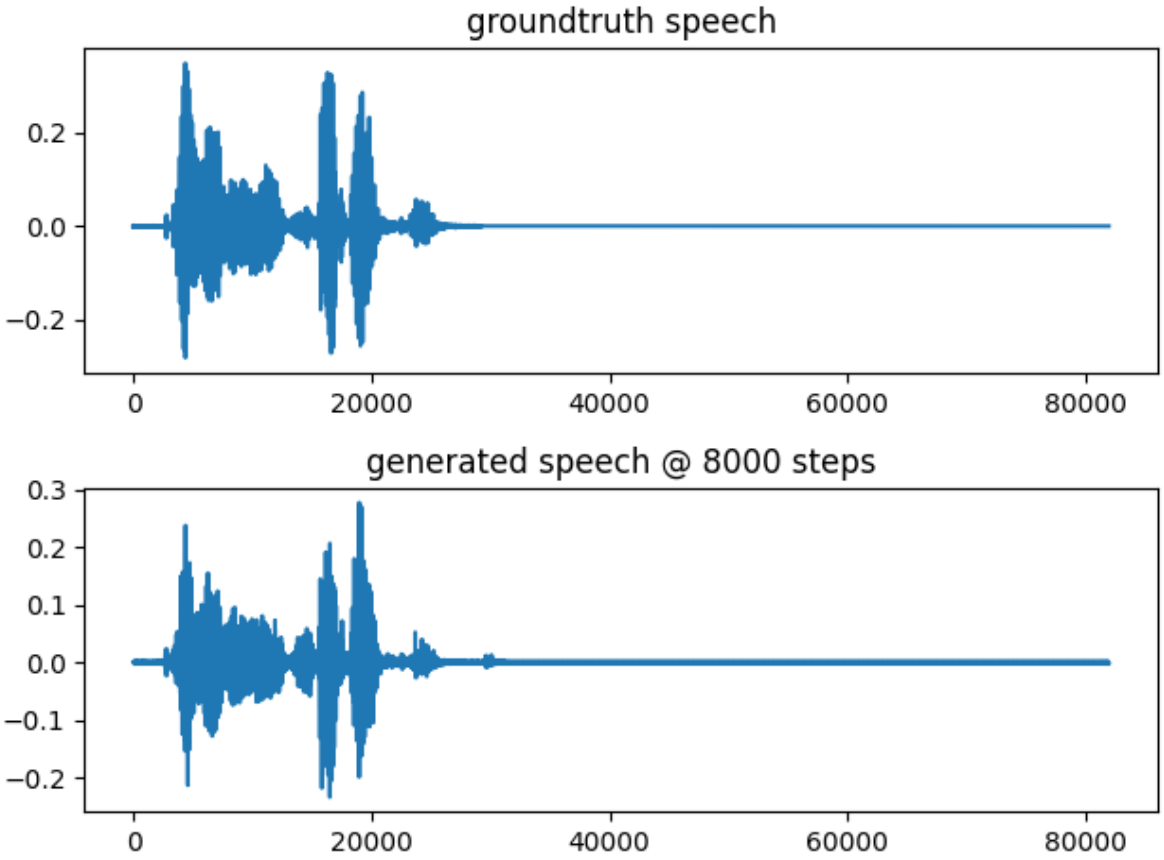
Mimo wielu godzin spędzonych podczas uczenia modeli, aby generowały mowę polską, nie udało się stworzyć satysfakcjonującej pary modeli. Taki wynik spowodowały między innymi następujące problemy:

- Niewystarczająca moc obliczeniowa – dostępny sprzęt w zbyt wolny sposób wykonywał proces uczenia (porównując do liczby iteracji gotowych modeli, na posiadanym sprzęcie trwałoby to kilka tygodni). Również próba uczenia w środowisku Google Colab okazała się bezskuteczna (środowisko jest darmowe, dlatego każdy użytkownik może z niego korzystać kilka godzin dziennie, po tym czasie dane są usuwane).
- Mały dataset – nagrane 164 pliki audio nie wystarczają, aby model „zapamiętał" wymowę wszystkich polskich wyrazów. Dla porównania, zbiór audio, z którego skorzystano do modelu z języku angielskim [6] posiada 13 100 plików o łącznej długości około 24 godzin.



Rysunek powyżej przedstawia wynik działania modelu FastSpeech2 po 20000 krokach działania. Z rysunku można wywnioskować, że generowany spektrogram daleko odbiega od użyteczności.

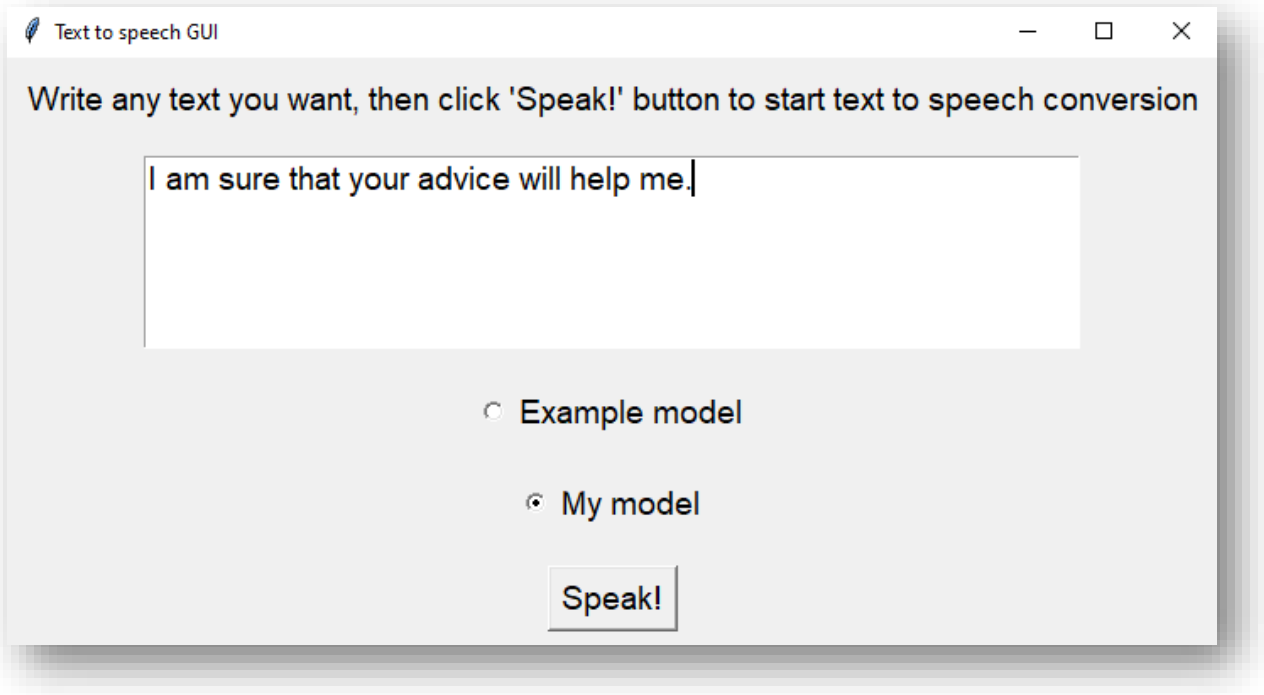
Autorowi udało się jednak wyuczyć model HiFi-GAN (8000 kroków), który potrafi zamienić spektrogram mel na dźwięk, który pozwala zrozumieć wyrazy.



Rysunek powyżej przedstawia wynik działania modelu HiFi-GAN. Jak można zauważyć, wynikowy sygnał jest znacznie zbliżony do oryginalnego. Potwierdza to również odsłuchanie plików audio, dostępnych w załączniku.

5. Końcowa aplikacja

W celu spełnienia założeń projektu, czyli stworzenia aplikacji pozwalającej na translację tekstu na mowę, autor postanowił wykorzystać wcześniej wytrenowany model FastSpeech 2 do generowania spektrogramów mel z tekstu oraz umieścić dla porównania możliwość połączenia go z własnym modelem HiFi-GAN oraz wcześniej wytrenowanym MelGAN.



W pole tekstowe należy wpisać tekst w języku angielskim, po czym wybrać odpowiedni model i nacisnąć przycisk „Speak!”.

6. Wnioski

Tworzenie własnych modeli sieci neuronowych nie należy do zadań łatwych. Pomimo dostępu do sporej wiedzy w Internecie, praktyczne jej wykorzystanie wymaga sporo wysiłku i pracy, a niekoniecznie musi dać oczekiwane rezultaty. Niniejszy projekt pokazał, że przed rozpoczęciem pracy należy dokładnie poznać wymagania przede wszystkim sprzętowe oraz realnie określić możliwości tworzenia dataset’u.

Źródła

1. https://pytorch.org/hub/nvidia_deeplearningexamples_tacotron2/
2. https://github.com/TensorSpeech/TensorFlowTTS
3. Yi Ren , Chenxu Hu , Xu Tan , Tao Qin , Sheng Zhao , Zhou Zhao , Tie-Yan Liu, *"FASTSPEECH 2: FAST AND HIGH-QUALITY END-TOEND TEXT TO SPEECH",* https://arxiv.org/pdf/2006.04558.pdf
4. Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, Aaron Courville*, "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis",* https://arxiv.org/pdf/1910.06711.pdf
5. Jungil Kong, Jaehyeon Kim, Jaekyoung Bae, *"HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis",* https://arxiv.org/abs/2010.05646
6. https://keithito.com/LJ-Speech-Dataset/