VOICE CLONING PROJECT

BY: R Mohan Poornachandra

Motivation and Use Cases

In this era of rising influence of Artificial Intelligence, a voice cloning model plays a very important role in personalizing, education, content creation and many such areas.

Voice Cloning is a great tool in industries like :

- Voice Assistants
- Dubbing and Audiobooks
- Education

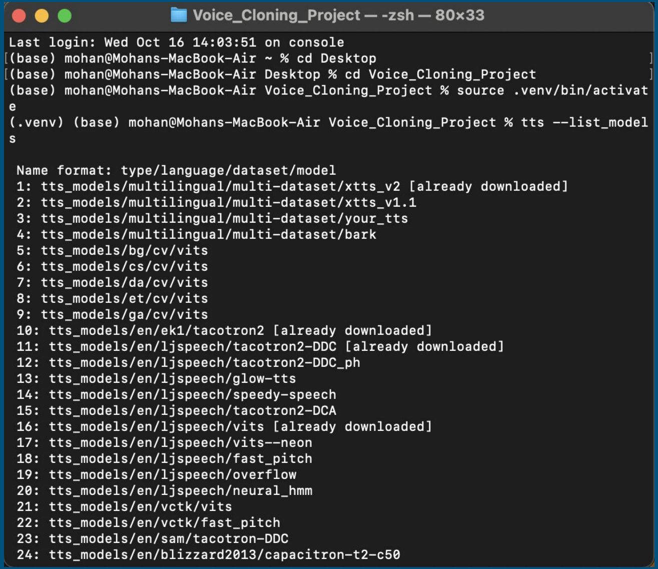
Installation Process

Installation was done in terminal and it various variables had to be given attention. The steps I followed to install TTS in my machine are:

1. Installing a compatible Python version. Python 3.9 , 3.10 or 3.11
2. Create a virtual environment say .venv and create a folder.
3. Change the current working directory to the new folder.
4. Run the command "pip install TTS"
5. Or we can also clone the coqui TTS repository and install all the requirements. (Command : " git clone https://github.com/coqui-ai/TTS"

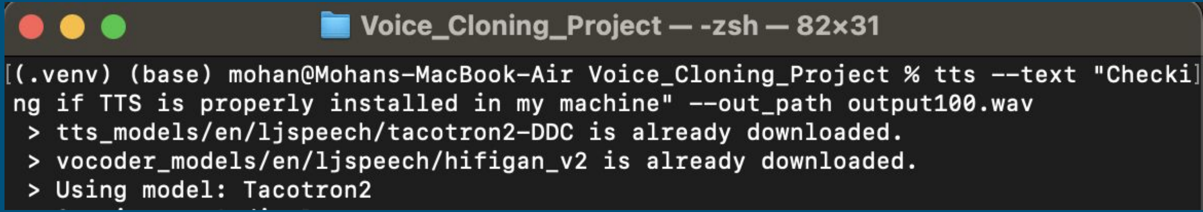
Installation Process Verification

These two commands should run without showing any error.



Voice_Cloning_Project — -zsh — 80×33

```
Last login: Wed Oct 16 14:03:51 on console
[(base) mohan@Mohans-MacBook-Air ~ % cd Desktop
[(base) mohan@Mohans-MacBook-Air Desktop % cd Voice_Cloning_Project
(base) mohan@Mohans-MacBook-Air Voice_Cloning_Project % source .venv/bin/activat
e
(.venv) (base) mohan@Mohans-MacBook-Air Voice_Cloning_Project % tts --list_model
s

 Name format: type/language/dataset/model
 1: tts_models/multilingual/multi-dataset/xtts_v2 [already downloaded]
 2: tts_models/multilingual/multi-dataset/xtts_v1.1
 3: tts_models/multilingual/multi-dataset/your_tts
 4: tts_models/multilingual/multi-dataset/bark
 5: tts_models/bg/cv/vits
 6: tts_models/cs/cv/vits
 7: tts_models/da/cv/vits
 8: tts_models/et/cv/vits
 9: tts_models/ga/cv/vits
 10: tts_models/en/ek1/tacotron2 [already downloaded]
 11: tts_models/en/ljspeech/tacotron2-DDC [already downloaded]
 12: tts_models/en/ljspeech/tacotron2-DDC_ph
 13: tts_models/en/ljspeech/glow-tts
 14: tts_models/en/ljspeech/speedy-speech
 15: tts_models/en/ljspeech/tacotron2-DCA
 16: tts_models/en/ljspeech/vits [already downloaded]
 17: tts_models/en/ljspeech/vits--neon
 18: tts_models/en/ljspeech/fast_pitch
 19: tts_models/en/ljspeech/overflow
 20: tts_models/en/ljspeech/neural_hmm
 21: tts_models/en/vctk/vits
 22: tts_models/en/vctk/fast_pitch
 23: tts_models/en/sam/tacotron-DDC
 24: tts_models/en/blizzard2013/capacitron-t2-c50
```

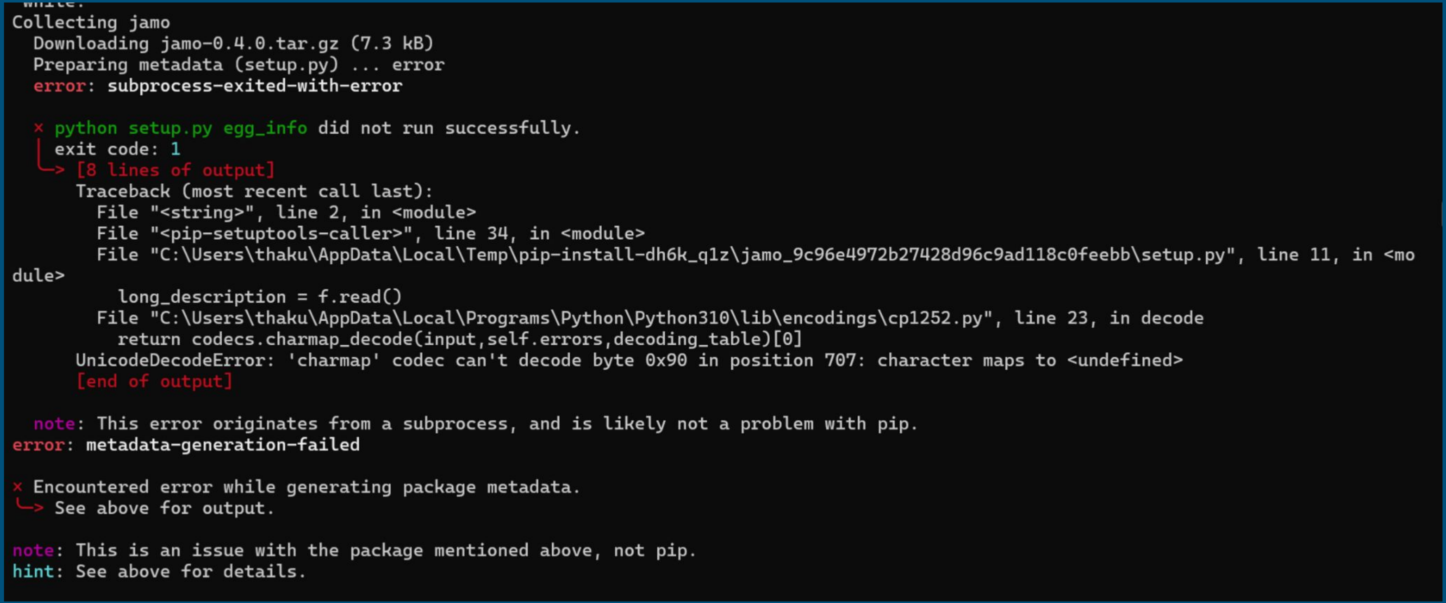


Voice_Cloning_Project — -zsh — 82×31

```
[(.venv) (base) mohan@Mohans-MacBook-Air Voice_Cloning_Project % tts --text "Checki
ng if TTS is properly installed in my machine" --out_path output100.wav
 > tts_models/en/ljspeech/tacotron2-DDC is already downloaded.
 > vocoder_models/en/ljspeech/hifigan_v2 is already downloaded.
 > Using model: Tacotron2
```

The above command should generate a wav file with the name output100.wav and it should say the text written in double quotes.

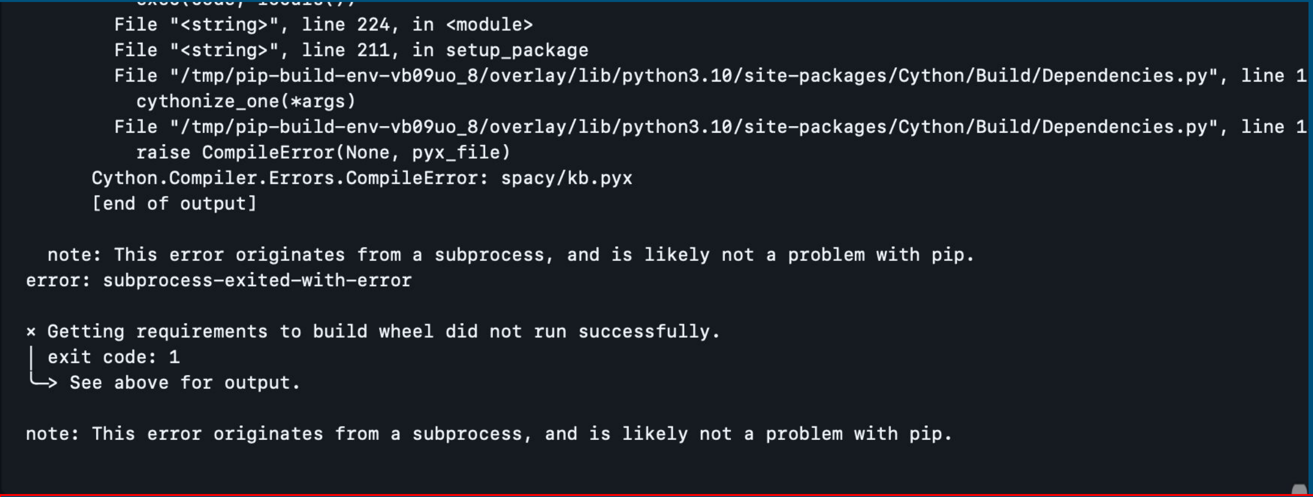
Possible issues while installing



```
Collecting jamo
  Downloading jamo-0.4.0.tar.gz (7.3 kB)
  Preparing metadata (setup.py) ... error
  error: subprocess-exited-with-error

  × python setup.py egg_info did not run successfully.
  │ exit code: 1
  ╰─> [8 lines of output]
      Traceback (most recent call last):
        File "<string>", line 2, in <module>
        File "<pip-setuptools-caller>", line 34, in <module>
        File "C:\Users\thaku\AppData\Local\Temp\pip-install-dh6k_q1z\jamo_9c96e4972b27428d96c9ad118c0feebb\setup.py", line 11, in <mo
dule>
          long_description = f.read()
        File "C:\Users\thaku\AppData\Local\Programs\Python\Python310\lib\encodings\cp1252.py", line 23, in decode
          return codecs.charmap_decode(input,self.errors,decoding_table)[0]
      UnicodeDecodeError: 'charmap' codec can't decode byte 0x90 in position 707: character maps to <undefined>
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
error: metadata-generation-failed

× Encountered error while generating package metadata.
╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
```

8 lines of output error: This error mainly arises due to incompatible Python version. I tried and changed my Python version to solve this.

Possible issues while installing



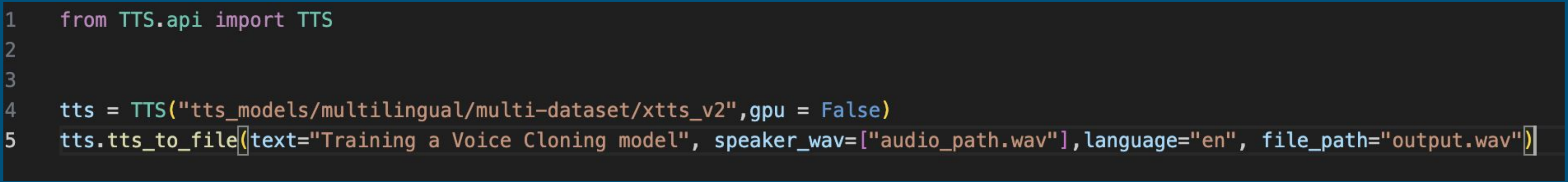
```
        File "<string>", line 224, in <module>
        File "<string>", line 211, in setup_package
        File "/tmp/pip-build-env-vb09uo_8/overlay/lib/python3.10/site-packages/Cython/Build/Dependencies.py", line 1
          cythonize_one(*args)
        File "/tmp/pip-build-env-vb09uo_8/overlay/lib/python3.10/site-packages/Cython/Build/Dependencies.py", line 1
          raise CompileError(None, pyx_file)
      Cython.Compiler.Errors.CompileError: spacy/kb.pyx
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
error: subprocess-exited-with-error

× Getting requirements to build wheel did not run successfully.
│ exit code: 1
╰─> See above for output.

note: This error originates from a subprocess, and is likely not a problem with pip.
```

Cython compile error / Spacy dependencies conflicts : This was a common error which troubled me for a long time. Tried a lot of solutions, but nothing worked. Then with the help of a GitHub issue, I was able to install a new spaCy version 3.7.5. The issue link: https://github.com/explosion/spaCy/issues/13449#issuecomment-2100827973

Developing an audio Dataset

It is very crucial to develop a good dataset of audio recordings with minimal background noise and high quality audio. The entire training is dependant on the quality of the audio and how expressive the speaker is. It is essential to make sure that the audio covers different words, phonetics, numbers and exclamations. This helps the model train on a wide spectrum of data.

Training the model: Code 1



```python
from TTS.api import TTS


tts = TTS("tts_models/multilingual/multi-dataset/xtts_v2",gpu = False)
tts.tts_to_file(text="Training a Voice Cloning model", speaker_wav=["audio_path.wav"],language="en", file_path="output.wav")
```

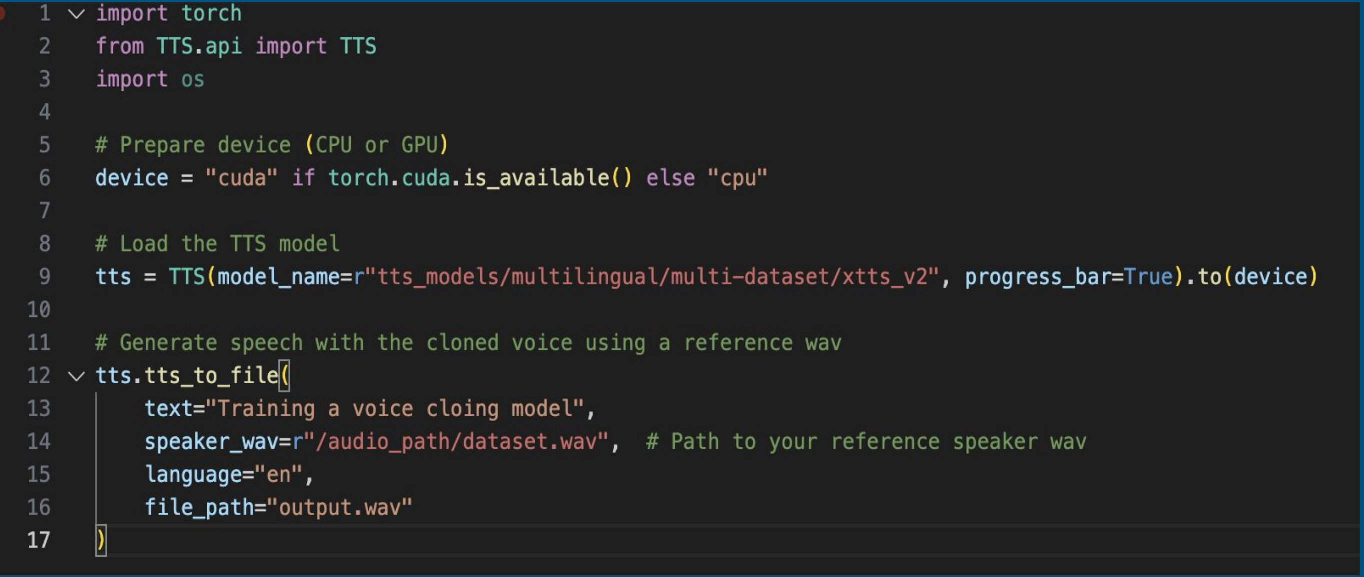
This code is very simple and was taken from YouTube. It doesn’t use GPU of the computer and runs only with the help of CPU. It can be trained by a single audio file and there is no need for a transcription.

Results :

https://drive.google.com/drive/folders/1Cnf7dYFlIDCsn01oyg_ZqKmbxsdy-j9e?usp=share_link

(Please download the files to play them)

Training the model: Code 2

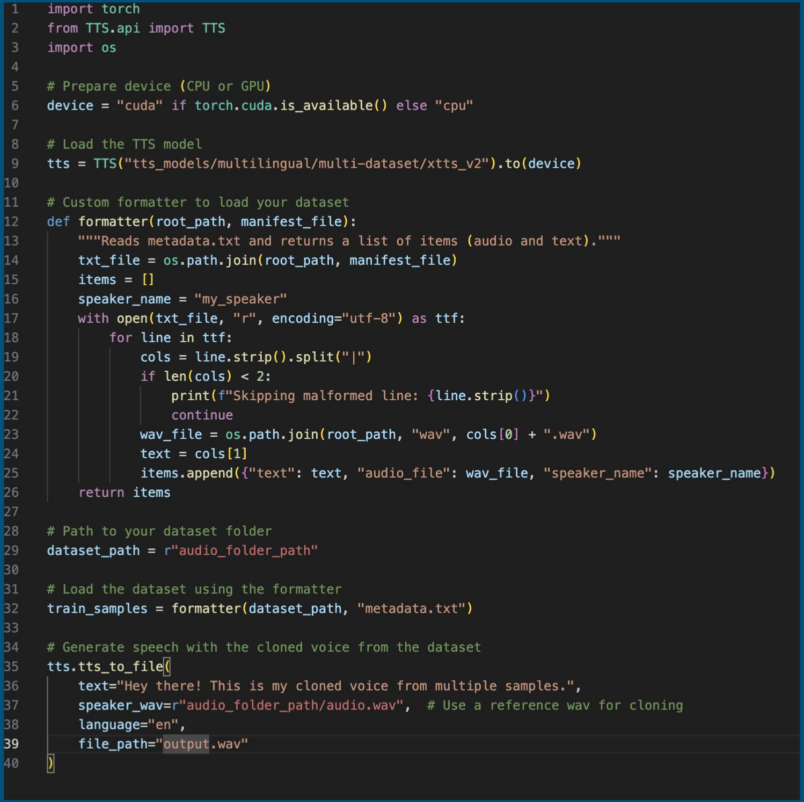


```python
import torch
from TTS.api import TTS
import os

# Prepare device (CPU or GPU)
device = "cuda" if torch.cuda.is_available() else "cpu"

# Load the TTS model
tts = TTS(model_name=r"tts_models/multilingual/multi-dataset/xtts_v2", progress_bar=True).to(device)

# Generate speech with the cloned voice using a reference wav
tts.tts_to_file(
    text="Training a voice cloing model",
    speaker_wav=r"/audio_path/dataset.wav",  # Path to your reference speaker wav
    language="en",
    file_path="output.wav"
)
```

This code also doesn’t require multiple audio, or any formatting.

Results :
https://drive.google.com/drive/folders/1Ig6PbkjtIp3IVjgOUtMMhTJsebdyS2cw?usp=share_link

Training the model: Code 3



```python
import torch
from TTS.api import TTS
import os

# Prepare device (CPU or GPU)
device = "cuda" if torch.cuda.is_available() else "cpu"

# Load the TTS model
tts = TTS("tts_models/multilingual/multi-dataset/xtts_v2").to(device)

# Custom formatter to load your dataset
def formatter(root_path, manifest_file):
    """Reads metadata.txt and returns a list of items (audio and text)."""
    txt_file = os.path.join(root_path, manifest_file)
    items = []
    speaker_name = "my_speaker"
    with open(txt_file, "r", encoding="utf-8") as ttf:
        for line in ttf:
            cols = line.strip().split("|")
            if len(cols) < 2:
                print(f"Skipping malformed line: {line.strip()}")
                continue
            wav_file = os.path.join(root_path, "wav", cols[0] + ".wav")
            text = cols[1]
            items.append({"text": text, "audio_file": wav_file, "speaker_name": speaker_name})
    return items

# Path to your dataset folder
dataset_path = r"audio_folder_path"

# Load the dataset using the formatter
train_samples = formatter(dataset_path, "metadata.txt")

# Generate speech with the cloned voice from the dataset
tts.tts_to_file(
    text="Hey there! This is my cloned voice from multiple samples.",
    speaker_wav=r"audio_folder_path/audio.wav",  # Use a reference wav for cloning
    language="en",
    file_path="output.wav"
)
```

Formatting the dataset:

https://docs.coqui.ai/en/latest/formatting_your_dataset.html

NOTE: Use only one audio file for this.

Results :

https://drive.google.com/drive/folders/1TNUSXlJo95pZJFT82lUnXUFw9LNaxjJX?usp=share_link

Fairness and Ethics of Voice Cloning models

These voice cloning models are very much powerful and can be used in a wrong way if given in wrong hands. So it is important to make sure that there is some software or security feature that protects and ensures the authenticity of the audio.

The challenge lies in creating a balance where this technology can be used for the good it promises, without infringing on privacy, creating unfair advantages, or being exploited for harmful purposes. Developers, users, and policymakers must work together to ensure that voice cloning is used ethically, with fairness at its core.

Future applications of my project

In the coming future, I would like to improve the quality of the output with better datasets and make it sound natural keeping the ethical constraints in my mind. I plan to use this project to leverage content creation, build some assistant or virtual character/agent.

Excited to learn more and apply this project in the coming future.