## **🔹 Original Setup Attempt**

* Project root:
* D:\ml\_projects\rvc\_project\
* You created a **venv** and installed RVC WebUI dependencies.
* You recorded your dataset (myrecording.wav) → split into clips (clip0.wav, clip1.wav, …).
* Stored here:
* D:\ml\_projects\rvc\_project\data\voice\trimmed\
* Normalized audio using ffmpeg-normalize.
* Tried to preprocess with:
* python infer-web.py --preprocess --dataset ...
* → but **errors occurred** because your repo (Retrieval-based-Voice-Conversion-WebUI) does not include CLI training scripts like preprocess.py or extract\_f0.py.
* Installed a lot of missing deps manually (future, decorator, pooch, soundfile, audioread, etc.) → got messy.

## **🔹 Problem Found**

* The repo you used was **WebUI-only**.
* Missing CLI scripts caused confusion.
* Dependency conflicts (FastAPI, Gradio, librosa, omegaconf) wasted time.

## **🔹 Clean Solution (Switch to Applio-RVC-Fork)**

We decided to move to:

D:\ml\_projects\Applio-RVC-Fork\

Why?

* Applio has **both WebUI + CLI helpers**.
* Dependencies are tested for **Windows + Python 3.10**.
* Much smoother than patching the old repo.

# **Final Setup (Step-by-Step)**

## **📂 Folder Structure**

D:\ml\_projects\

├── rvc\_project\ # Your old attempt (keep as backup)

│ └── data\voice\trimmed\ # Your existing dataset

│

└── Applio-RVC-Fork\ # New clean repo (recommended)

├── venv\ # Python 3.10 virtual environment

├── assets\dataset\myvoice\ # Copy of your trimmed dataset

├── logs\myvoice\ # Preprocessed + extracted features

└── infer-web.py # Main entry (WebUI + CLI)

## **⚙️ Step Sequence**

1. **Clone repo**
2. git clone https://github.com/SayanoAI/Applio-RVC-Fork.git
3. cd Applio-RVC-Fork
4. **Create Python 3.10 venv**
5. py -3.10 -m venv venv
6. venv\Scripts\activate
7. **Install dependencies**
8. python -m pip install "pip<24.1"
9. pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
10. pip install -r requirements.txt
11. **Copy dataset**
12. mkdir assets\dataset\myvoice
13. xcopy /E /I D:\ml\_projects\rvc\_project\data\voice\trimmed assets\dataset\myvoice
14. **Preprocess + Extract f0 (via CLI)**
15. python infer-web.py --is\_cli
    * Choose **Preprocess Dataset** → assets\dataset\myvoice
    * Choose **Extract f0 features** → assets\dataset\myvoice
16. **Train the model**
17. python train.py -e myvoice -sr 40k -bs 4 -g 0
    * -e myvoice = experiment name
    * -sr 40k = 40kHz sample rate (lighter for 4GB GPU)
    * -bs 4 = batch size 4 (safe for GTX 1650 VRAM)
    * -g 0 = GPU ID
18. **Run WebUI for inference**
19. python infer-web.py
20. → opens http://127.0.0.1:7865 in browser.

# **✅ Achieved**

* Recorded your dataset (myrecording.wav).
* Cleaned & split into clips (clip0.wav → clip81.wav).
* Normalized + prepared dataset (trimmed folder).
* Identified issues with original repo.
* Switched to **Applio-RVC-Fork** for stable CLI + UI.
* Clear plan for: preprocess → f0 extraction → training → inference.

👉 This workflow ensures you can reproduce training anytime with **zero dependency chaos**.

RVC Voice Training Setup Summary

# 📂 Folder Structure

D:\ml\_projects\  
 ├── rvc\_project\ # Your old attempt (keep as backup)  
 │ └── data\voice\trimmed\ # Your existing dataset  
 │  
 └── Applio-RVC-Fork\ # New clean repo (recommended)  
 ├── venv\ # Python 3.10 virtual environment  
 ├── assets\dataset\myvoice\ # Copy of your trimmed dataset  
 ├── logs\myvoice\ # Preprocessed + extracted features  
 └── infer-web.py # Main entry (WebUI + CLI)

# ⚙️ Step Sequence

1. Clone repo:  
    git clone https://github.com/SayanoAI/Applio-RVC-Fork.git  
    cd Applio-RVC-Fork
2. Create Python 3.10 venv:  
    py -3.10 -m venv venv  
    venv\Scripts\activate
3. Install dependencies:  
    python -m pip install "pip<24.1"  
    pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118  
    pip install -r requirements.txt
4. Copy dataset:  
    mkdir assets\dataset\myvoice  
    xcopy /E /I D:\ml\_projects\rvc\_project\data\voice\trimmed assets\dataset\myvoice
5. Preprocess + Extract f0 (via CLI):  
    python infer-web.py --is\_cli  
    - Choose Preprocess Dataset → assets\dataset\myvoice  
    - Choose Extract f0 features → assets\dataset\myvoice
6. Train the model:  
    python train.py -e myvoice -sr 40k -bs 4 -g 0  
    -e myvoice = experiment name  
    -sr 40k = 40kHz sample rate (lighter for 4GB GPU)  
    -bs 4 = batch size 4 (safe for GTX 1650 VRAM)  
    -g 0 = GPU ID
7. Run WebUI for inference:  
    python infer-web.py  
    → opens http://127.0.0.1:7865 in browser.

# ✅ Achieved

- Recorded your dataset (myrecording.wav)  
- Cleaned & split into clips (clip0.wav → clip81.wav)  
- Normalized + prepared dataset (trimmed folder)  
- Identified issues with original repo  
- Switched to Applio-RVC-Fork for stable CLI + UI  
- Clear plan for: preprocess → f0 extraction → training → inference