# Voice Cloning (8 GB RAM)

This guide walks through recording your voice, cloning it with RVC, synthesizing speech, and animating a photo with SadTalker — all on a PC with ~8 GB memory. We emphasize lightweight settings (e.g. smaller models, lower resolutions) to fit your 8 GB hardware.

## Prerequisites

* **Hardware:** A CUDA-compatible GPU with ~8 GB VRAM (e.g. NVIDIA RTX 2060). RVC recommends at least 8 GB VRAM for training[[1]](https://applio-page.fandom.com/wiki/RVC_(Retrieval-Voice-Conversion)#:~:text=The%20minimum%20required%20for%20local,compatible%20with%20Nvidia%20graphics%20cards). If your GPU or RAM is limited, work with shorter clips or smaller models (see below).
* **Software:** Python 3.8+, Git, and FFmpeg must be installed. Use a virtual environment (e.g. Conda or python -m venv) to isolate dependencies. We’ll install libraries (Torch, etc.) per tool. Ensure no heavy programs are running in the background to conserve memory.

## Step 1: Record Your Voice Dataset

* **Script & Recording:** Choose a phonetically-rich text (news article, book excerpt, etc.) and record yourself reading it in a quiet room. Use a decent microphone and keep your volume consistent. Aim for at least 10–15 minutes of clean speech (more is better; 30+ minutes if possible)[[2]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/wiki/FAQ-(Frequently-Asked-Questions)#:~:text=A%20dataset%20of%20around%2010min,to%2050min%20is%20recommended). Shorter recordings (5–10 min) can work if quality is high, but 10+ min is recommended for good results[[2]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/wiki/FAQ-(Frequently-Asked-Questions)#:~:text=A%20dataset%20of%20around%2010min,to%2050min%20is%20recommended).
* **File Format:** Export recordings as WAV (16-bit PCM, 44100 Hz). To ease processing, split your audio into clips ~5–10 s long (e.g. clip1.wav, clip2.wav, …). Store all WAV files in one folder (e.g. dataset/voice/).

## Step 2: Train the Voice Cloning Model (RVC)

We use the RVC WebUI to train a model of **your voice**. RVC works with a small dataset (>=10 min) and even *“can train quickly on relatively modest GPUs”*[[3]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=,RMVPE%E6%A0%B9%E7%BB%9D%E5%93%91%E9%9F%B3%E9%97%AE%E9%A2%98%E3%80%82%E6%95%88%E6%9E%9C%E6%9C%80%E5%A5%BD%EF%BC%88%E6%98%BE%E8%91%97%E5%9C%B0%EF%BC%89%E4%BD%86%E6%AF%94crepe_full%E6%9B%B4%E5%BF%AB%E3%80%81%E8%B5%84%E6%BA%90%E5%8D%A0%E7%94%A8%E6%9B%B4%E5%B0%8F%20%2A%20A%E5%8D%A1I%E5%8D%A1%E5%8A%A0%E9%80%9F%E6%94%AF%E6%8C%81).

1. **Clone and install RVC:** Open a terminal and run:

* git clone https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI.git  
  cd Retrieval-based-Voice-Conversion-WebUI  
  python -m venv venv # create a virtual environment
* Activate it (venv\Scripts\activate on Windows or source venv/bin/activate on Linux/Mac). Then install PyTorch and RVC’s requirements. For 8 GB GPUs, use a compatible PyTorch (CUDA) version and small settings:
* pip install torch torchvision torchaudio # install PyTorch  
  pip install -r requirements.txt # install RVC dependencies
* (Follow PyTorch’s official [install guide](https://pytorch.org/get-started/locally/) to pick the right CUDA version.)  
  [[4]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=pip%20install%20torch%20torchvision%20torchaudio)[[5]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=)

1. **Prepare voice data:** Create a folder inside the RVC project to hold your clips, for example:

* mkdir data/user\_voice
* Copy all your WAV clips into this folder. Make sure they are reasonably loud and noise-free. (RVC’s FAQ notes that 10–50 minutes of clean speech gives the best results[[2]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/wiki/FAQ-(Frequently-Asked-Questions)#:~:text=A%20dataset%20of%20around%2010min,to%2050min%20is%20recommended).)

1. **Launch RVC WebUI:** Run:

* python infer-web.py
* This opens the RVC interface in your browser[[6]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=python%20infer). In the “Train Voice Model” section, select your data/user\_voice/ folder as the training data. Start training with default settings. (By default RVC uses a pretrained base model and fine-tunes it; it can learn from even 10+ min of data[[3]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=,RMVPE%E6%A0%B9%E7%BB%9D%E5%93%91%E9%9F%B3%E9%97%AE%E9%A2%98%E3%80%82%E6%95%88%E6%9E%9C%E6%9C%80%E5%A5%BD%EF%BC%88%E6%98%BE%E8%91%97%E5%9C%B0%EF%BC%89%E4%BD%86%E6%AF%94crepe_full%E6%9B%B4%E5%BF%AB%E3%80%81%E8%B5%84%E6%BA%90%E5%8D%A0%E7%94%A8%E6%9B%B4%E5%B0%8F%20%2A%20A%E5%8D%A1I%E5%8D%A1%E5%8A%A0%E9%80%9F%E6%94%AF%E6%8C%81)[[2]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/wiki/FAQ-(Frequently-Asked-Questions)#:~:text=A%20dataset%20of%20around%2010min,to%2050min%20is%20recommended).) **Tip (8 GB):** If you run into CUDA OOM errors, try reducing batch size (set it to 1) or use fewer clips at once[[7]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/wiki/FAQ-(Frequently-Asked-Questions)#:~:text=For%20training%2C%20reduce%20the%20batch,cards%20still%20have%20a%20chance). Training may take a few hours depending on data size; the WebUI will save a model file (e.g. G\_\* .pth) when done.

1. **Build voice index:** Once trained, in the RVC UI click **“Build Index”** for your model. This finalizes it for faster conversion.

## Step 3: Generate Speech Audio (TTS)

Now we synthesize speech from text, then convert it into your cloned voice. We’ll use a lightweight TTS and then RVC.

* **Option A – Bark (Suno.ai):** Bark is a powerful TTS that can generate natural speech. *Caution:* Bark’s full model needs ~12 GB VRAM[[8]](https://github.com/suno-ai/bark#:~:text=The%20full%20version%20of%20Bark,SUNO_USE_SMALL_MODELS%3DTrue). On 8 GB, set the flag SUNO\_USE\_SMALL\_MODELS=True before running, which loads smaller sub-models[[8]](https://github.com/suno-ai/bark#:~:text=The%20full%20version%20of%20Bark,SUNO_USE_SMALL_MODELS%3DTrue). Install and run Bark:
* pip install git+https://github.com/suno-ai/bark.git
* In Python:
* from bark import generate\_audio, preload\_models  
  preload\_models()  
  text = "Hello, this is a test sentence."  
  audio\_array = generate\_audio(text)
* This gives a NumPy waveform. Save it to WAV (e.g. using scipy.io.wavfile.write("speech.wav", rate, audio\_array)). *Note:* Bark **does not** clone your voice; it uses generic voices. But it’s very natural. If Bark is too heavy on memory, skip to Option B.  
  [[8]](https://github.com/suno-ai/bark#:~:text=The%20full%20version%20of%20Bark,SUNO_USE_SMALL_MODELS%3DTrue)[[9]](https://github.com/suno-ai/bark#:~:text=pip%20install%20git%2Bhttps%3A%2F%2Fgithub.com%2Fsuno)
* **Option B – Tortoise TTS (JarodMica):** Tortoise supports voice conditioning (more advanced) and may run on less GPU RAM but needs CPU/GPU setup. Clone Jarod’s repo and install:
* git clone https://github.com/JarodMica/ai-voice-cloning.git  
  cd ai-voice-cloning  
  ./setup-cuda.sh # Linux: installs dependencies (or run setup-cuda.bat on Windows)  
  ./start.bat # launches the Tortoise GUI/CLI
* (These scripts install models and libraries[[10]](https://github.com/JarodMica/ai-voice-cloning#:~:text=1). Ensure Python 3.11 is installed as required.) In the Tortoise interface, enter your text and use *your recorded voice samples* as references. Tortoise will generate an output WAV (e.g. tortoise\_output.wav) that sounds like your voice.
* **Convert to your cloned voice (RVC):** If you used Bark (or any other TTS), you’ll now convert that speech into *your* voice using RVC. In the RVC WebUI, go to **“Inference – Convert”**. Set *Source Audio* to the generic speech WAV (speech.wav or tortoise\_output.wav), and select your trained voice model. Click **Convert**; RVC will produce a new WAV (e.g. speech\_converted.wav) that says the same text in *your* voice (using the voice clone).

Now you have speech\_converted.wav – the text spoken in your cloned voice.

**References:** Official docs and forums were used for commands and best practices. For example, RVC’s GitHub shows pip installs and WebUI usage[[4]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=pip%20install%20torch%20torchvision%20torchaudio)[[6]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=python%20infer), Bark’s repo notes its VRAM needs[[8]](https://github.com/suno-ai/bark#:~:text=The%20full%20version%20of%20Bark,SUNO_USE_SMALL_MODELS%3DTrue), and SadTalker’s documentation covers setup and settings[[11]](https://github.com/OpenTalker/SadTalker#:~:text=conda%20create%20)[[14]](https://learn.thinkdiffusion.com/sadtalker-talking-head-videos/#:~:text=Face%20model%20resolution%20and%20GFPGAN,face%20enhancer). These ensure the above instructions are accurate for low-memory environments.

[[1]](https://applio-page.fandom.com/wiki/RVC_(Retrieval-Voice-Conversion)#:~:text=The%20minimum%20required%20for%20local,compatible%20with%20Nvidia%20graphics%20cards) RVC (Retrieval-Voice-Conversion) | Applio Wiki | Fandom

<https://applio-page.fandom.com/wiki/RVC_(Retrieval-Voice-Conversion)>

[[2]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/wiki/FAQ-(Frequently-Asked-Questions)#:~:text=A%20dataset%20of%20around%2010min,to%2050min%20is%20recommended) [[7]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/wiki/FAQ-(Frequently-Asked-Questions)#:~:text=For%20training%2C%20reduce%20the%20batch,cards%20still%20have%20a%20chance) FAQ (Frequently Asked Questions) · RVC-Project/Retrieval-based-Voice-Conversion-WebUI Wiki · GitHub

<https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/wiki/FAQ-(Frequently-Asked-Questions)>

[[3]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=,RMVPE%E6%A0%B9%E7%BB%9D%E5%93%91%E9%9F%B3%E9%97%AE%E9%A2%98%E3%80%82%E6%95%88%E6%9E%9C%E6%9C%80%E5%A5%BD%EF%BC%88%E6%98%BE%E8%91%97%E5%9C%B0%EF%BC%89%E4%BD%86%E6%AF%94crepe_full%E6%9B%B4%E5%BF%AB%E3%80%81%E8%B5%84%E6%BA%90%E5%8D%A0%E7%94%A8%E6%9B%B4%E5%B0%8F%20%2A%20A%E5%8D%A1I%E5%8D%A1%E5%8A%A0%E9%80%9F%E6%94%AF%E6%8C%81) [[4]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=pip%20install%20torch%20torchvision%20torchaudio) [[5]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=) [[6]](https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI#:~:text=python%20infer) GitHub - RVC-Project/Retrieval-based-Voice-Conversion-WebUI: Easily train a good VC model with voice data <= 10 mins!

<https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI>

[[8]](https://github.com/suno-ai/bark#:~:text=The%20full%20version%20of%20Bark,SUNO_USE_SMALL_MODELS%3DTrue) [[9]](https://github.com/suno-ai/bark#:~:text=pip%20install%20git%2Bhttps%3A%2F%2Fgithub.com%2Fsuno) GitHub - suno-ai/bark: Text-Prompted Generative Audio Model

<https://github.com/suno-ai/bark>

[[10]](https://github.com/JarodMica/ai-voice-cloning#:~:text=1) GitHub - JarodMica/ai-voice-cloning

<https://github.com/JarodMica/ai-voice-cloning>