[Hello Security]

WGUiSW

0x01 Say Hello to Security

- AI & Voice Cloning

Krystian Bajno, 2024

_baycode.eu

[Table of Contents]

0x01 Say Hello To Security – AI & Voice Cloning

0x02 Whoami

0x03 Links section

0x04 Exploring online

0x05 Exploring the world of self-hosted

0x06 Anatomy of a deepfake

0x07 Practical steps to perform voice cloning

0x08 SO VITS SVC

0x09 RVC

0x0A Threat modelling

0x0B What are possible AI uses – examples

0x0C Let us create a song!

0x0D Falling into a rabbit hole

0x0E Q&A

0x0F Thank you

[Links are clickable]

WGUiSW _baycode.eu

0x02 Whoami

Krystian Bajno

Cyber Security Specialist
Penetration Tester

Full-Stack Software Engineer
Backend, Frontend, Mobile

AKADEMIA WIT

Comp. Sci. I – Cloud Computing Technology

Comp. Sci. II - Cloud Computing Architecture and Security

0x03 Links section

Useful projects

https://github.com/voicepaw/so-vits-svc-fork
https://github.com/WadRex/RVCompact
https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI
https://github.com/litagin02/rvc-tts-webui
https://github.com/w-okada/voice-changer
https://github.com/facebookresearch/demucs
https://github.com/AUTOMATIC1111/stable-diffusion-webui
https://github.com/comfyanonymous/ComfyUI
https://github.com/lllyasviel/Fooocus
https://github.com/s0md3v/roop
https://github.com/guoyww/AnimateDiff
https://github.com/lllyasviel/ControlNet
https://github.com/facebookresearch/llama
https://huggingface.co/microsoft/phi-2
https://github.com/henrymaas/AudioSlicer
https://github.com/flutydeer/audio-slicer
https://github.com/coqui-ai/tts

Model repositories

https://huggingface.co
https://civitai.com

Useful links

https://machine-learning.paperspace.com/wiki/machine-learning-models-explained
https://www.hardware-corner.net/guides/computer-to-run-llama-ai-model/
https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/
https://colab.google/
https://paperswithcode.com/task/speech-synthesis/
https://paperswithcode.com/
https://openart.ai/workflows

Recommended AI YouTube channels

https://www.youtube.com/@Fireship
https://www.youtube.com/@NerdyRodent
https://www.youtube.com/@sentdex
https://www.youtube.com/@sedetweiler
https://www.youtube.com/@houseofdim
https://www.youtube.com/@sebastiankamph
https://www.youtube.com/@enigmatic_e
https://www.youtube.com/@OlivioSarikas
https://www.youtube.com/@EndangeredAI

0x04 Exploring online

The 2022/3 rise of Generative AI (GAN, LLM)

⭐ Selected speech synthesis providers

ElevenLabs

https://elevenlabs.io

- Polish company! 🇵🇱
- Text to Speech
- Speech to Speech
- Dubbing (video translator)
- Make custom models (paid)

HeyGen

https://www.heygen.com

- Lip sync!
- Creating whole digital avatars
- Translating videos
- Speech to Speech

Pros:

1. **Ease of use, cutting edge**
2. **No tech knowledge** requirement
3. **No hardware** requirement
4. **Support,** big funding
5. **API integrations**

Cons:

1. **Commercial,** limited free plans, pricy tier limitations, credit systems
2. **Potential political correctness** limitations
3. **No real-time voice streaming** (at the current moment, all it takes is WebRTC)
4. **Data collection**

Voice

https://www.resemble.ai

https://www.heygen.com

https://elevenlabs.io

Imagery

https://www.bing.com/search?q=Bing+AI&showconv=1

https://openai.com/dall-e-3

https://www.midjourney.com/

https://www.krea.ai/home

LLM

https://www.bing.com/search?q=Bing+AI&showconv=1

https://chat.openai.com/

https://bard.google.com/chat

NVIDIA
Microsoft
Google
Meta
Apple
Open Source Community

The biggest players in **AI** as of 2024.

0x05 Exploring the world of Self-Hosted

Let's do it offline and open source

Requirements

1. **For training**, **voice infering**, **images generation**, and **smaller LLMs** (such as phi-2 and LLAMA-7B, approx 10B parameters), **8 GB of GPU VRAM** is sufficient (eg. RTX 3060, or Apple M1 will do due to chip mixing RAM and VRAM, but M1 works 9x slower than RTX). For bigger LLM's, such as LLAMA-70B **- 52 GB of GPU VRAM** is a a bare minimum.
2. **Lots of free storage**.

Alternatively deploy cloud environment for training

Deploy model training on selected **Cloud Computing Platform**, train it in minutes, download the checkpoints, and kill the instance, just like spinning up and destroying a hash cracking machine. You can also use **Google Colab**. Bigger LLM's however require that much resources to **run**. It does not mean you need bigger LLMs to get good results.

Pros:

1. **Free**
2. **Huge model base**
3. **Huge amount of software**
4. **Full control**
5. **Limitless possibilities**
6. **Cutting edge**

Cons:

1. **Requires a bit of tech knowledge**
2. **Requires good GPU**
3. **Only community support**

Voice

https://github.com/voicepaw/so-vits-svc-fork
https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI
https://github.com/litagin02/rvc-tts-webui
https://github.com/w-okada/voice-changer
https://github.com/facebookresearch/demucs
https://github.com/flutydeer/audio-slicer
https://github.com/coqui-ai/tts

Imagery

https://github.com/comfyanonymous/ComfyUI **(strongly recommended)**
https://github.com/AUTOMATIC1111/stable-diffusion-webui
https://github.com/lllyasviel/Fooocus **(love the simplicity)**
https://github.com/s0md3v/roop

LLM

https://github.com/facebookresearch/llama
https://huggingface.co/microsoft/phi-2

Model repositories

https://huggingface.co
https://civitai.com/

Useful links

https://paperswithcode.com/

0x06 Anatomy of a deepfake

Unsupervised deep-learning nature of Generative Adversarial Networks

Picture:

1. Find **one**, good picture of the impersonated person.
2. Substitute the face on a selected picture with impersonated person face (eg. using **Roop, Fooocus, ComfyUI**), or generate a completely fake image with substituted face.
3. Upscale the picture (for example using **Stable Diffusion**, **R-ESRGAN** upscaler).

Video:

1. Find a picture of the impersonated person (one is all it takes).
2. Substitute the face on all frames of the video, or generate completely fake video using **ComfyUI.**
3. Upscale all frames of the video.
4. Voice clone the impersonated person and substitute your own voice in the video with a clone.

Mind your ethics. Everything is possible, but what for?
Ask the person you clone for approval.

0x07 Practical steps to perform voice cloning

Make me sound like you

1. **Acquire a 10 or more minutes sample** of the impersonated person voice.
2. **Cut the sample into approx. 5 second pieces** and remove noise, silence (using **AudioSlicer, audacity**), this sample will be later used for GAN unsupervised model training and **no data labeling** is needed.
3. **Train the model using RVC or SO-VITS-SVC** for 200 epochs or more. (Can take a few hours). Increasing epochs too much will not improve the quality due to ***overfitting***.
4. **Use the created model for inference** or **real-time inference streaming.**

The **RVC** and **SVC** models are not interchangeable and need to be trained separately.

Make sure to create a **loopback interface** or use a **mixer** in order to avoid audio feedback while streaming in **real-time.**

Streaming voice is easier than a complete deepfake video real-time streaming due to resources extensive use.
Expect the voice streaming lag to be 0.3 of a second, whereas processing video takes way more time. In order to stream-infer voice + video, you will need to use cloud computing platform or a powerful computer. It is needed to match the audio/video latencies, yet it is perfectly doable in 2024.

Mind your ethics. Everything is possible, but what for?
Ask the person you clone for approval.

0x08 SO VITS SVC

Step by step

Activate Python environment

```
.\venv\Scripts\Activate.ps1 - Windows
source ./venv/Scripts/activate – Linux, MacOS
```

1. Place dataset files created with AudioSlicer into

`dataset_raw/<speaker_id>/**/<files.wav>`

2. Pre-process the dataset

```
svc pre-resample - converts your audio to mono 44.1khz files
svc pre-config - downloads a few configuration files and puts them in the correct directory.
```

3. Modify number of epochs and batch size

in `logs/44k/config.json` to match **VRAM** or training will crash.

4. Continue pre-processing to optimize to "crepe" prediction method

```
svc pre-hubert -fm crepe - downloads and runs a speech model pre-training.
```

5. Train the model (this could take a few hours)

Model checkpoints will be available in `logs/44k - G_x.pth, D_x.pth`

```
svc train –t
```

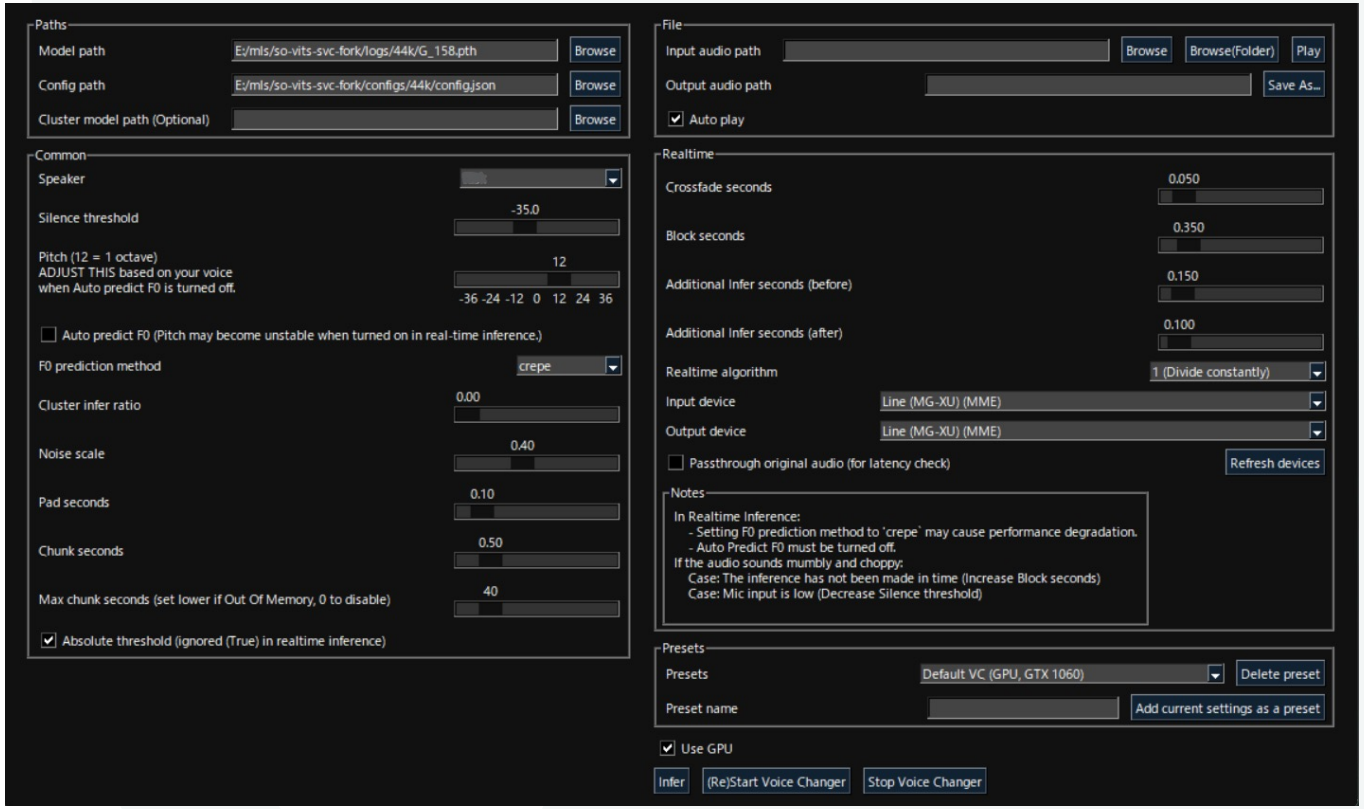
6. Use the model

```
svc gui
```

Usage:

Select the **Model path**, **Config path**, and the **Speaker.**

1. To **infer file to file**, select file input audio path, output audio path and press **Infer**.
2. To **change your voice in real-time** - select input and output device, disable auto pitch prediction, adjust Pitch to your voice, press (Re)/Start Voice Changer, and use the output device as input in selected application (eg. Discord, Teams, Telegram). **Make sure to use a loopback interface and a set of headphones to prevent audio feedback.**



0x09 RVC

Step by step

Activate Python environment and launch the web-ui.

`.\venv\Scripts\Activate.ps1` - Windows

`source ./venv/Scripts/activate` – Linux, MacOS

Web interface will be available at http://localhost:7897

1. Name the experiment (project)

2a. Load and pre-process the dataset in the user interface

RVC can split and denoise audio, but you can specify a directory with pre-processed files.

2b. Extract the pitch and perform HuBERT pre-training technique

3. Train the model and create feature index

a) **One-Click training** (Recommended)

This will perform step 2b, model training, and feature index creation **automatically**.

b) **Train model button** – train the model manually after step 2b.

c) **Train feature index button** – create feature index manually.

Model checkpoints and index will be saved in logs directory. If you want to share the model, you'll need to extract and compile the .pth checkpoint from logs directory using **ckpt processing** tab, export should be about **60 MBs** in size. The model that is ready to use should reside in assets/weights directory.

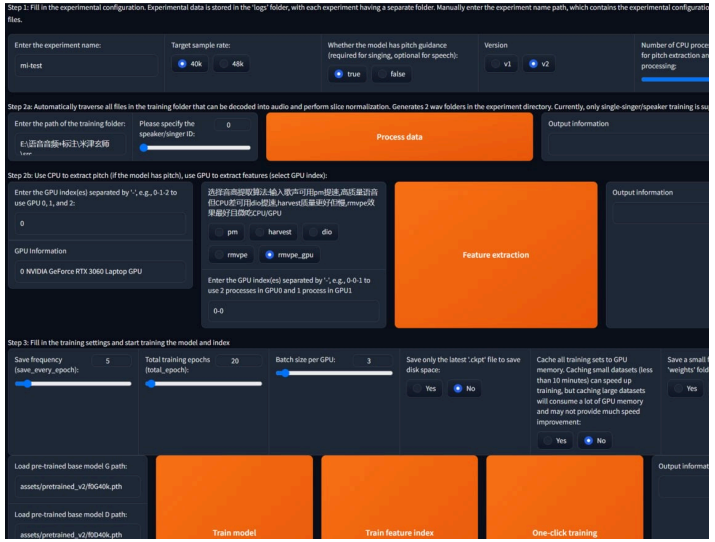
4. Use the model

There are two user interfaces available

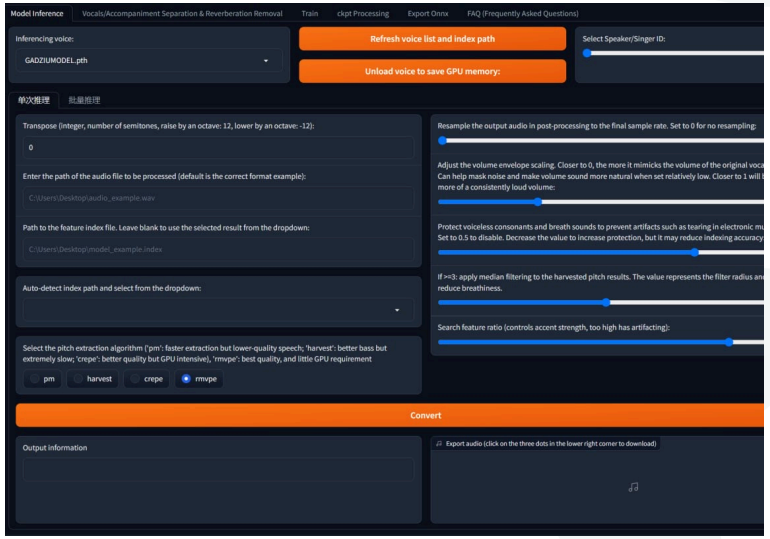
a) For **inference** stay in the web-ui

b) For **real-time streaming** launch gui_v1.py

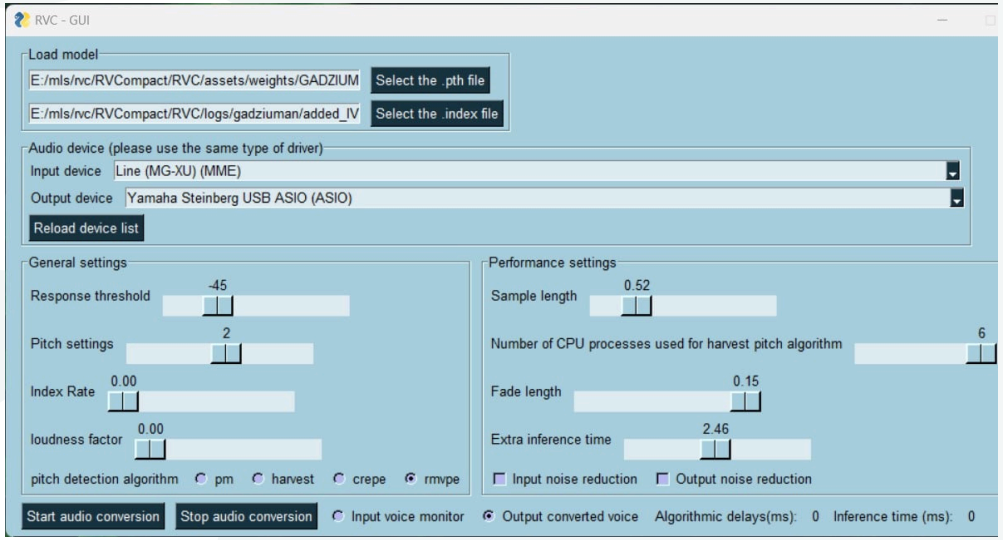
1. Training view



2. Inference view



3. Real-Time inference view



0x0A Threat modelling

⚠️ The wide emerging threat of 2024

Due to making AI accessible to wide-public, the new threats have emerged.

What/Who could be the possible target?

- **Anyone** who gets his voice cloned
- **Anyone** who gets his face cloned
- Applications with **bio-authentication** (voice)
- Publicly speaking people
- YouTubers
- Influencers
- Politicians
- The internet

What is the threat?

- AI **phishing**, **vishing**, **impersonation, identity theft**
- Bypassing **bio-authentication** (voice)
- **Blackmailing**
- **Disinformation** and increase of **trolling**
- Automated false identity **scamming** bots, **catfishing**
- Fake, bot generated **influencers**
- Increase of AI generated **explicit** pictures
- Abuse of people voice and **copyrights**
- Overreliance on Artificial Intelligence could have potential **negative impact on cognitive abilities** of specialists.
- Flood of LLM generated **spam** content over the internet
- Growing concerns about the **reliability** of digital data due to **AI hallucinations** and **impersonation. Anything digital can be generated.**

0x0B What are possible AI uses - examples

Do good, not evil

Automation

Creating bots that auto-respond to clients, leveraging LLMs, text-to-speech, and GAN networks for voice inference.

Parsing big sets of data

LLMs can parse big sets of data in order to give useful insights, or even provide code solutions.

Security Assessments

Useful in social-engineering, red-teaming, and comprehensive phishing assessments.

Art creation

Creating art – pictures, animations, songs.

Breaking the language barrier

Translating, dubbing the video/audio into various languages in order to reach wider audience.

Sock Puppets

Creating sock-puppet false identities in order to achieve anonymity and infiltrate criminal communities is easier than ever.

0x0C Let us create a song!

Let’s do it now

1. **Clone** the voice of a friend using **RVC** or **SO-VITS**.
2. **Download** the song to exchange the voice in.
3. **Demux** the singer, bass, drums, guitars and pianos on origin using **Demucs.**
4. **Infer** the singer voice using the cloned model.
5. **Reassemble** the song in **audacity** or any other **DAW**.
6. **Enjoy** making your friend a rockstar.

0x0D Falling into a rabbit hole

Win a bunny

1. **What cipher** is unbreakable provided it is used correctly - and only once, with the key length of plaintext?
2. **What** is steganography?
3. **What web vulnerability** is number one in OWASP 2021?
4. **Imagine** – there is a banking trojan that replaces the addresses during transfers in client browser memory. How can you secure the ***client*** web application?

0x0E Q&A

Ask me anything you want

0x0F Thank you

Presentation in PDF format is available on **https://news.baycode.eu**

*Meet me again at **https://wguisw.org***