European Commission

South-Eastern Finland University of Applied Sciences

Transformative impact of AI on eArchiving processes

Anssi Jääskeläinen

Research manager Xamk/Digitalia

Anssi.jaaskelainen@xamk.fi

https://memorylab.fi/

Memory Lab

Digitalia

Agenda

- Short intro
- Virtualization & containers
- Hugging Face
  - “Treasury of AI”
- Hands on examples
  - OCR, ASR and TTS
    - +HTR bonus
  - Example dockers and instructions ~16Mb zip
    - https://tinyurl.com/earkdemo
- Q&A

Xamk / Digitalia / Memory Lab

- **Xamk - South-Eastern Finland University of Applied Sciences**
  - https://www.xamk.fi/en/frontpage/
- **Memory Lab**
  - AI specialized ~680 000€ technical environment
    - Hippu supercomputer
  - Operational June 23.
  - https://memorylab.fi
    - Now also in English ☺

- **Digitalia – Research Center on Digital Information Management**
  - Founded 2015
  - Usability of digital materials
  - Automating thingies
  - 3D - Visualization
  - AI development
  - Digitalia.fi
  - VR things: https://olddigitalia.memorylab.fi/

Virtualization & Containers

Virtualization

- Type 1 (bare metal)
  - Lightweight OS, runs directly on hardware
    - Proxmox, VMWare ESXi, HyperV, Xenserver
    - GPU passthrough, vgpu, mig etc. possibilities
- Type 2 (hosted)
  - Software layer on an OS
    - VMWare Workstation, VirtualBox
    - GPU utilization might not be possible
- Easy to migrate within Type, difficult from bare metal to hosted
- Why needed
  - Would you install questionable things on your computer?
  - extra security
  - host remains untouched
  - Efficiency
- Use if a full “computer” experience is needed

Containers

- Container uses host OS kernel and runs on isolated space
  - More efficient and starts faster
  - Ideal for pipelines and microservices
  - Portable solution
- Use if “only a runnable solution” is needed
- Things to consider
  - Containers as stateless
    - Restart removes all content
  - Occasional cleanup needed
  - Basic cli usage needed

Hugging Face

OCR: Optical Character Recognition

Ocrbench Leaderboard - a Hugging Face Space by echo840

PaddleOCR & training

- Open-source
- Small and fast
- Three different ML models inbuild:
  - Line detection
  - Line translation detection model
  - Text recognition model
- 80+ languages
- Good GPU support
- The original model had issues with recognizing diacritics

- Training: ~160 000 line images
  - Machine and handwritten & synthetic material
    - TRDG (TextRecognitionDataGenerator)
    - https://github.com/Belval/TextRecognitionDataGenerator

kallistui niin syvään, että sen äärimmäiset oksat hipoivat kirkasta

Results (CER) Character Error Rate

Model	Test 1 (4273 line images)	Test 2 (3475 line images)	Test 3 (3247 line images)	Test4 htr (714 line images)
Tesseract	4,6%	2,7%	4,4%	79,3%
Paddle (original)	6,7%	3,9%	6,8%	50,2%
Paddle (retrained)	2,0%	1,2%	2,3%	20,7%

About the training times

- Even though the model was/is small (few megabytes) → Be prepared with a GPU

Hardware	Training time	Notes
VM with 16 core CPU	~240 days	Estimated time given by the training
VM with 2 x A100	~1 day	
Container with 8 x A100	2 hours	

- So, should you buy A100 GPU card..? (6912 cuda cores / 80Gb nvram) or can you use consumer GPUs?
  - GTX 1080 (2560 cuda cores / 8-12Gb nvram)
  - RTX 3080 (8704 cuda cores / 10-16Gb nvram)
  - RTX A4000 (6144 cuda cores / 20Gb nvram)

Still interested?

- Trained model and API
  - https://github.com/project-AIDA/Finnish_PaddleOCR
- Training model for retraining
  - https://huggingface.co/Kansallisarkisto/PaddleOCR_training
- Dockerized version
  - https://github.com/xamkfi/digitalia-aida-extended-paddle-demo
- Public demo
  - https://memorylab.fi/AIDA/extended-paddle-demo/
    - This is in Finnish, but just upload a text image and press OCR-lue (OCR-read) button

ASR: Automatic Speech Recognition

https://huggingface.co/spaces/hf-audio/open_asr_leaderboard

ASR - practical approach

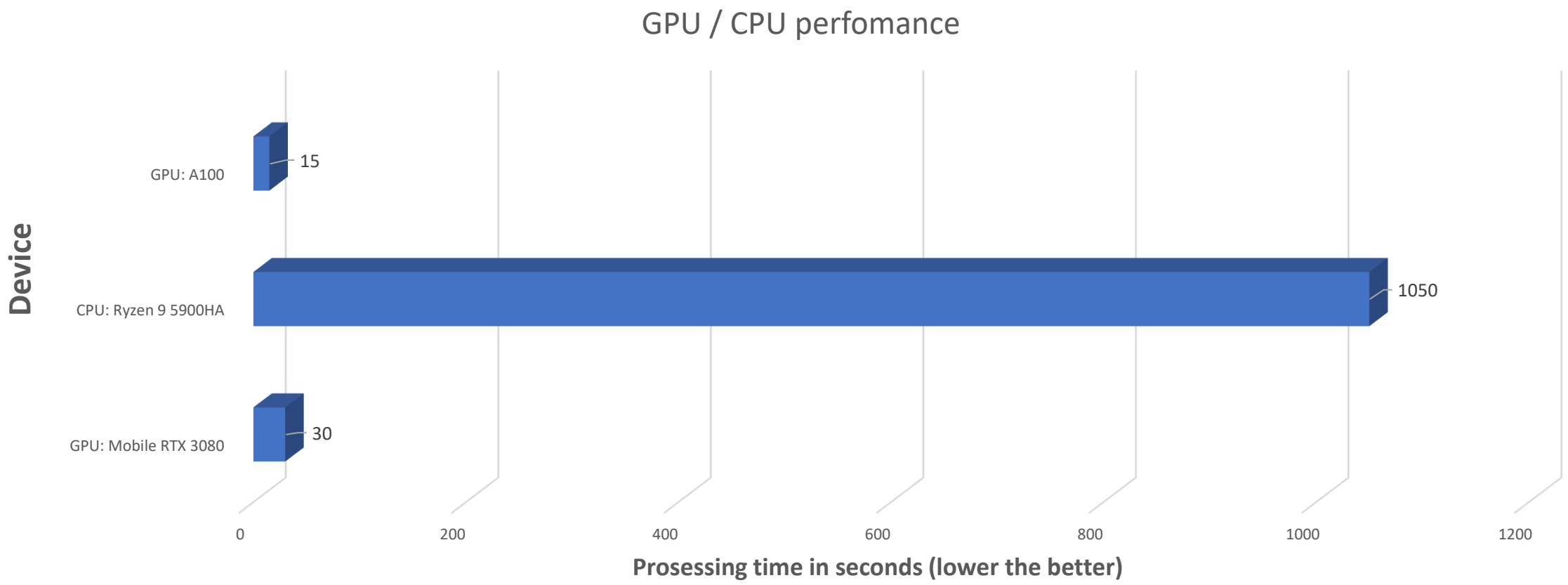
- Picking a model
- Enter https://huggingface.co/
- Use top search bar and find whisper-large-v3-turbo
  - Introduction
  - Usage
    - Examples, code, explanations
    - Use this model (button)
  - Right menu (finetunes)
    - https://huggingface.co/adriszmar/whisper-large-v3-turbo-es
    - WER Spanish 6,91 → 5,34
- Simplified taking into use with AI
  - Copilot / chatGPT / other *"I want to use this AI model (link) write me a Gradio code for this"*
- Some models are gated → sign in and create an access key

	Large-v3	Large-v3-turbo
Size	~25Gb	1,6Gb
parameters	1550M	809M
Average **WER**	7,44	7,83
RTFX	145	200

Whisper v3 large turbo

- Human robustness an accuracy on English (+others) speech recognition
- Transcribes ~100 different languages
- ~100 to English translation
  - The provided demo sticks with transcription
- can predict timestamps (sentence and word level)
- Uses FFmpeg library to load audio → supports all the file types FFmpeg supports.
  - →You can also process various video files
- Automatically handles longer audios (code)
  - Either by sequential or chunked approach
- https://memorylab.fi/demot/asr/

ASR speed


GPU / CPU perfomance
Device
GPU: A100
CPU: Ryzen 9 5900HA
GPU: Mobile RTX 3080
15
1050
30
0
200
400
600
800
1000
1200
Prosessing time in seconds (lower the better)

TTS: Text To Speech

TTS Spaces Arena - a Hugging Face Space by Pendrokar

What’s the point?

- Accessibility regulations (European Accessibility Act)
  - Archived digital documents, websites, and databases must be accessible to people with disabilities. This includes ensuring compatibility with screen readers and other assistive technologies
- “Modern users don’t read”, access to data needs to be quick, convenient and engaging

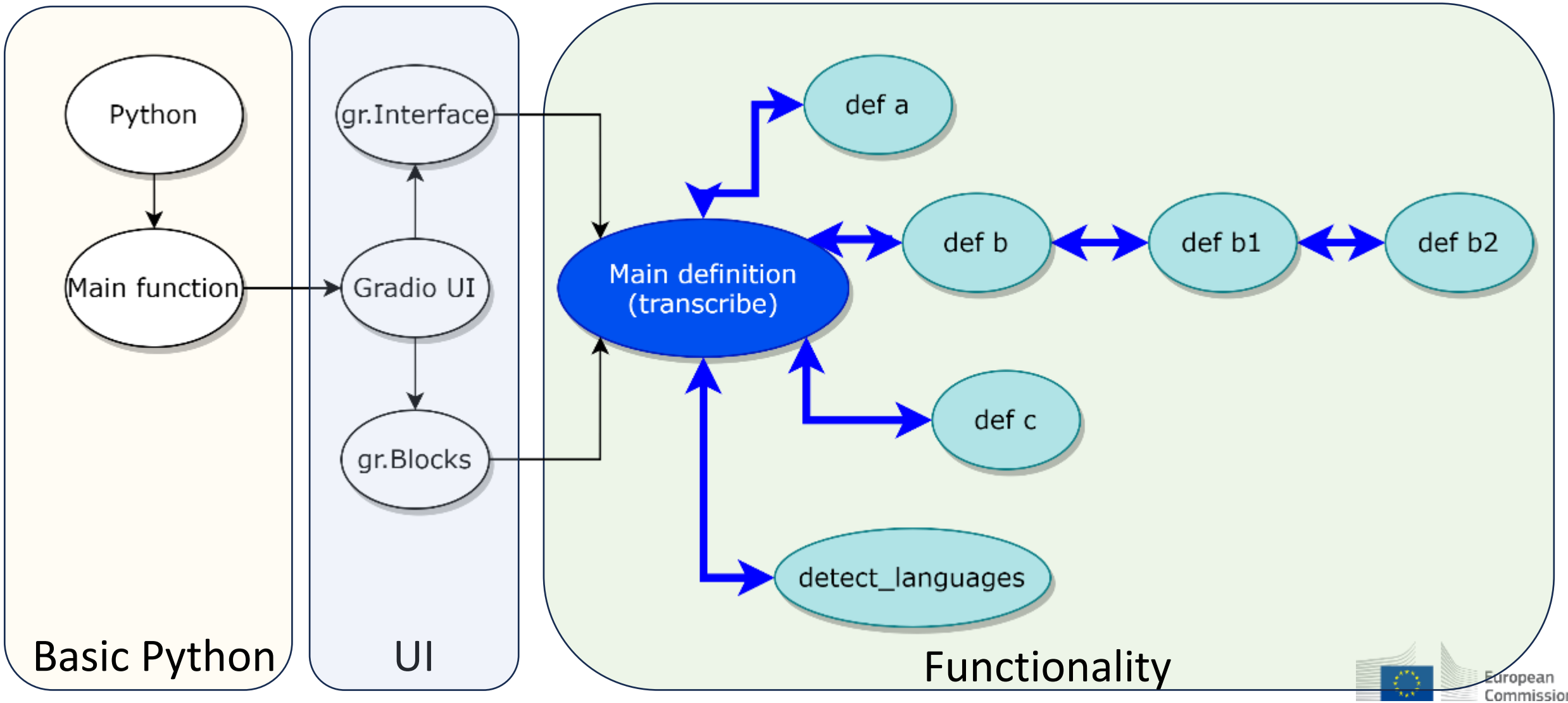
Kokoro model

- 82 million parameters
- Multiple voices available
- Several quantization options
- Public demo: https://memorylab.fi/demot/tts/

HTR

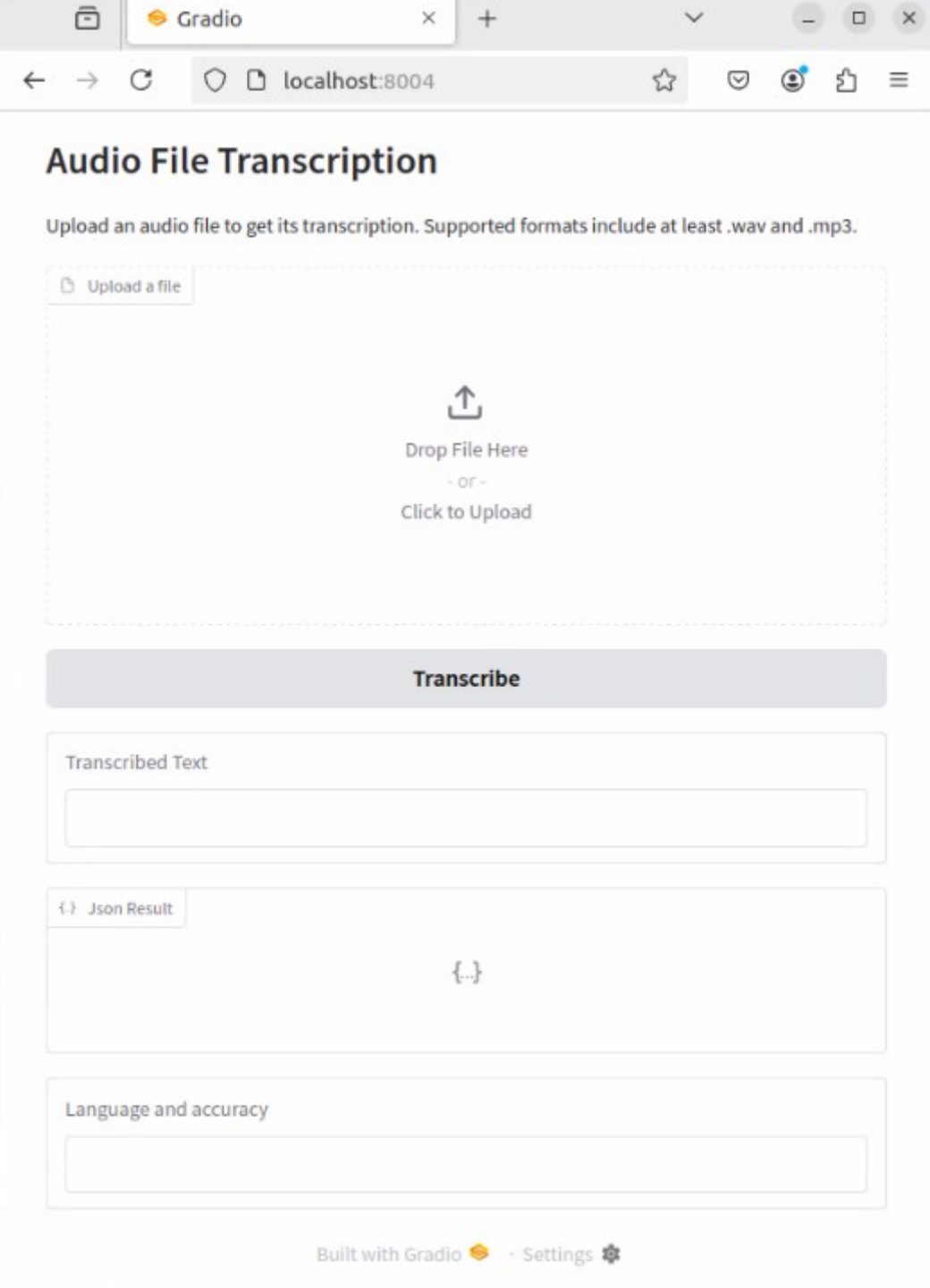
- If the container generation is finished and if time left

Basic of Python Gradio applications



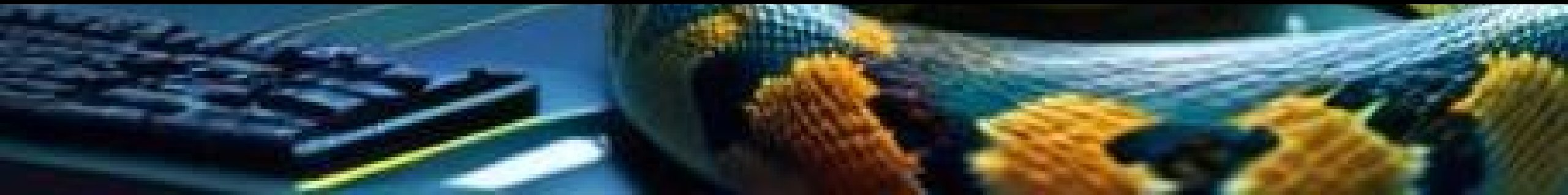
UI

```
with gr.Blocks() as demo:
    gr.Markdown("# Audio File Transcription\nUpload an audio file to get its transcription. Supported formats include at least .wav and .mp3.")
    file_input = gr.File(label="Upload a file")
    transcribe_button = gr.Button("Transcribe")
    text_output = gr.Textbox(label="Transcribed Text")
    result_output = gr.JSON(label="Json Result")
    accuracy_output = gr.Textbox(label="Language and accuracy")

    transcribe_button.click(fn=transcribe, inputs=file_input, outputs=[text_output, result_output, accuracy_output])

demo.launch(server_port=8004, server_name="0.0.0.0")
```



Things to be noted

- Some Hugging Face repos are gated
  - You need to login and create an access token: https://huggingface.co/settings/tokens
  - Connect your run environment with the token
- Python evolves over time, the latest isn't always the best for running AI
- Gradio should be installed with pip install, not with conda even if you are in conda env.
- Start with short audios / videos to see how your system performs
- If using AMD GPU instead of Nvidia/CUDA
  - Probably some code changes due to ROCm

Questions, comments, criticism, worries, etc.?

Anssi Jääskeläinen
Anssi.jaaskelainen@xamk.fi
https://memorylab.fi/