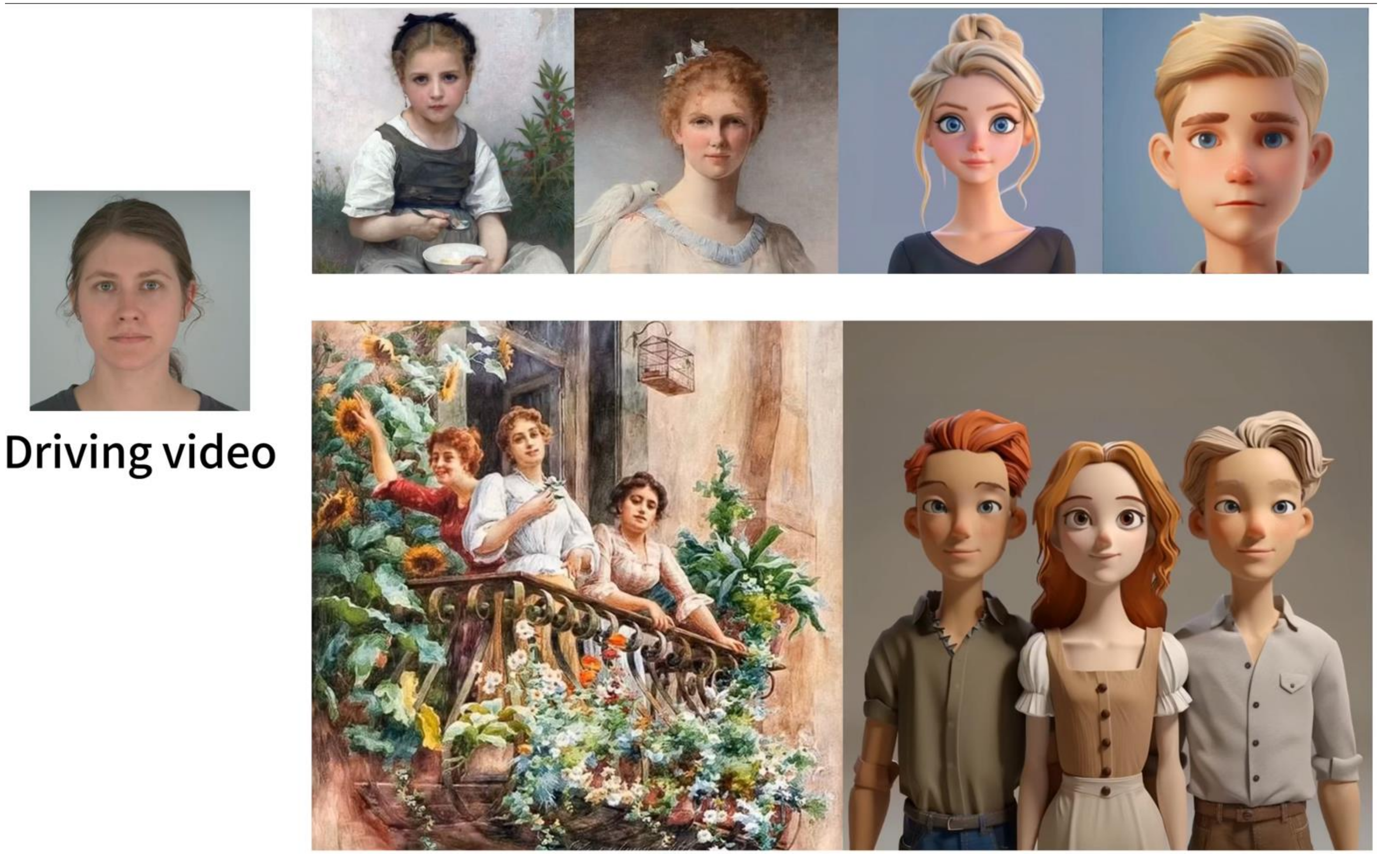
AI 활용 특강
LLM for ~~AI~~
~~Vtuber(3)~~

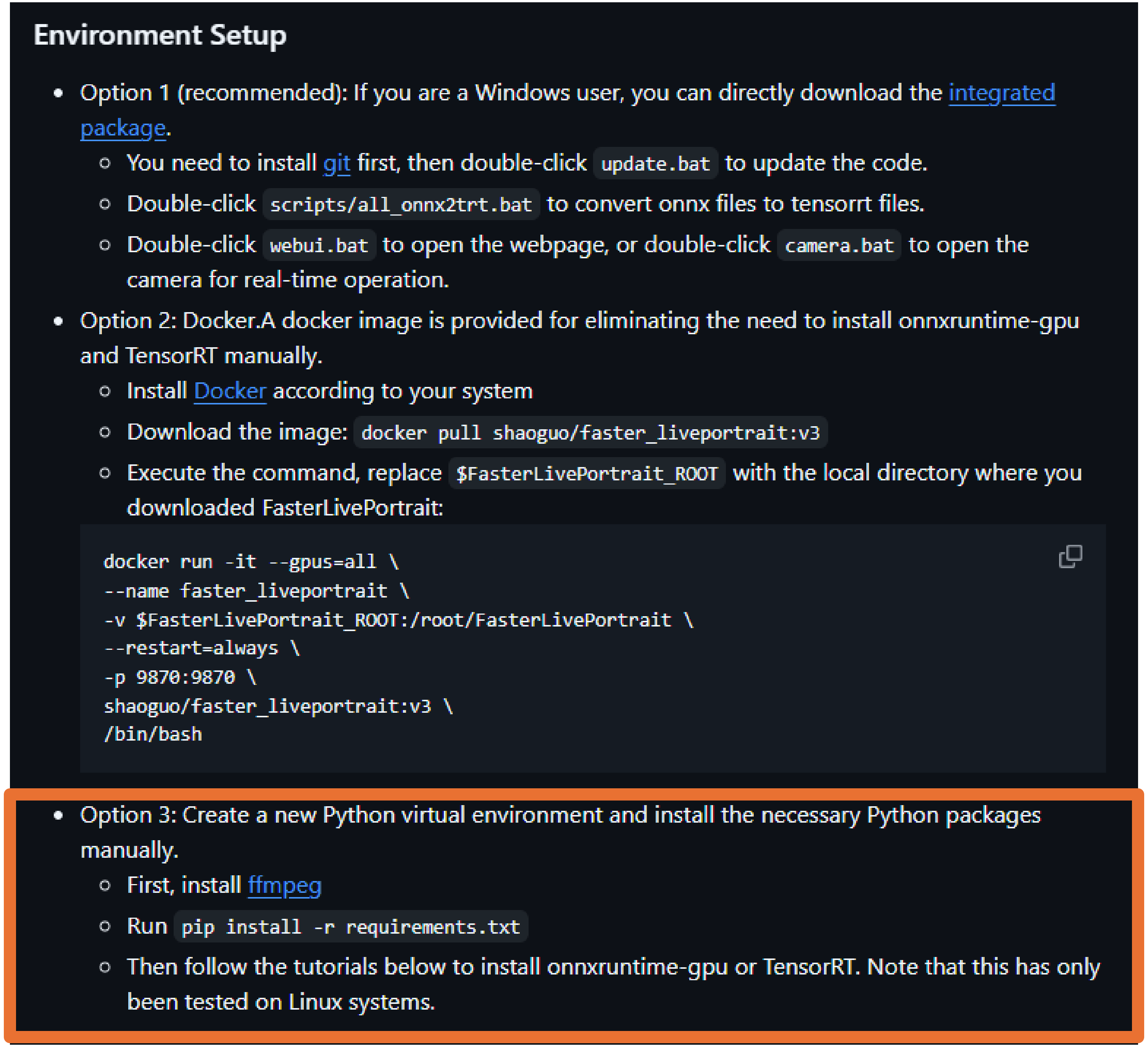
LivePortrait: Efficient Portrait Animation with Stitching and Retargeting Control



Live Portrait Animation From a Still Image

Setup guide

각자 편한 방법으로!



Environment Setup

- Option 1 (recommended): If you are a Windows user, you can directly download the integrated package.
    - You need to install git first, then double-click `update.bat` to update the code.
    - Double-click `scripts/all_onnx2trt.bat` to convert onnx files to tensorrt files.
    - Double-click `webui.bat` to open the webpage, or double-click `camera.bat` to open the camera for real-time operation.
- Option 2: Docker.A docker image is provided for eliminating the need to install onnxruntime-gpu and TensorRT manually.
    - Install Docker according to your system
    - Download the image: `docker pull shaoguo/faster_liveportrait:v3`
    - Execute the command, replace `$FasterLivePortrait_ROOT` with the local directory where you downloaded FasterLivePortrait:

```
docker run -it --gpus=all \
--name faster_liveportrait \
-v $FasterLivePortrait_ROOT:/root/FasterLivePortrait \
--restart=always \
-p 9870:9870 \
shaoguo/faster_liveportrait:v3 \
/bin/bash
```

- Option 3: Create a new Python virtual environment and install the necessary Python packages manually.
    - First, install ffmpeg
    - Run `pip install -r requirements.txt`
    - Then follow the tutorials below to install onnxruntime-gpu or TensorRT. Note that this has only been tested on Linux systems.

Prepare : install ffmpeg, Powershell

- Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1’))
- choco install ffmpeg

- 출처 : https://velog.io/@shinyejin0212/chocolatey-Windows용-패키지-매니저-chocolatey-설치-하기

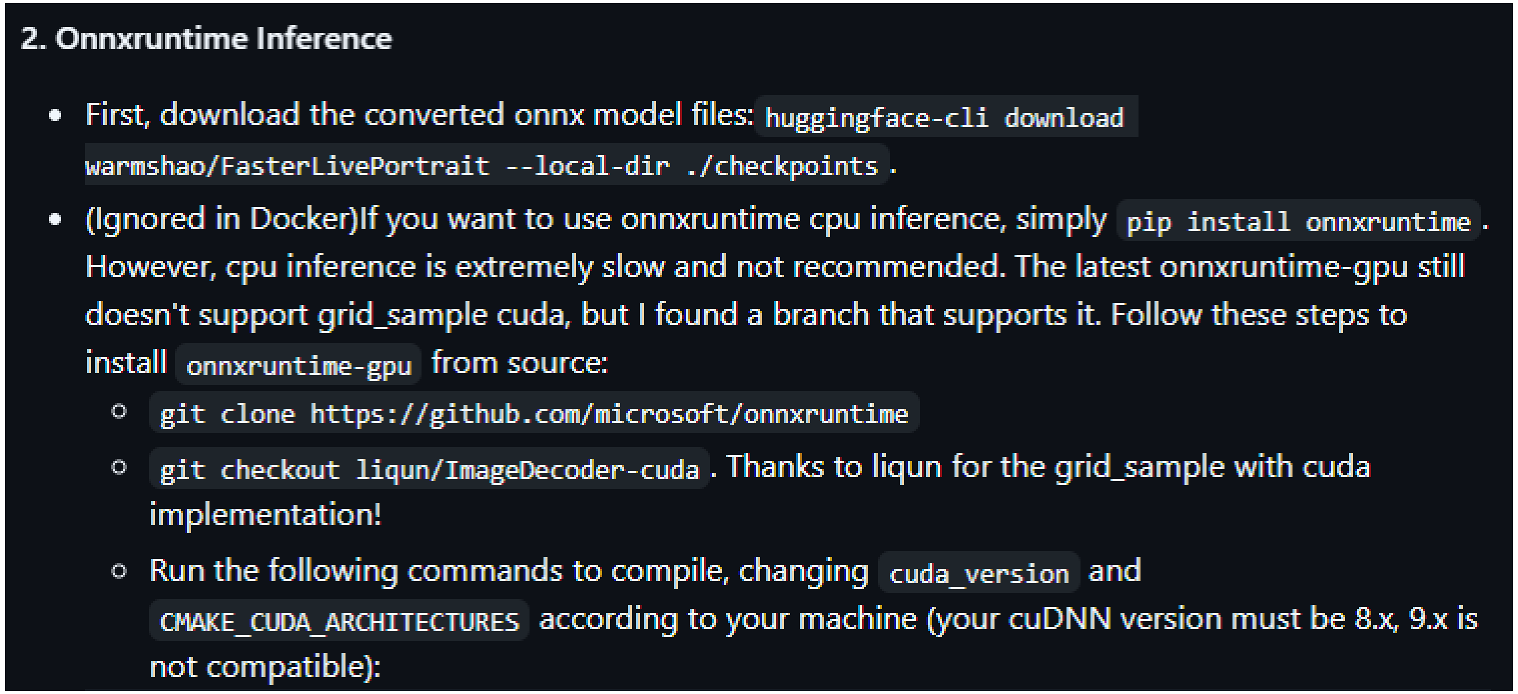
warmshao/**FasterLivePortrait**

- cd 저장할 경로
- git clone https://github.com/warmshao/FasterLivePortrait
- **Set-ExecutionPolicy Unrestricted (PowerShell 해당자)**
- python –m venv venv
- ./venv/Scripts/Activate.ps1
- python.exe –m pip install --upgrade pip
- pip install –r requirements_win.txt

- 이후로 huggingface-cli 설치 해야함
  - 참고사항 : https://huggingface.co/docs/huggingface_hub/main/en/guides/cli

Onnx 모델 추론하기 위한 설치

- huggingface-cli download warmshao/FasterLivePortrait --local-dir ./checkpoints
- pip install onnxruntime

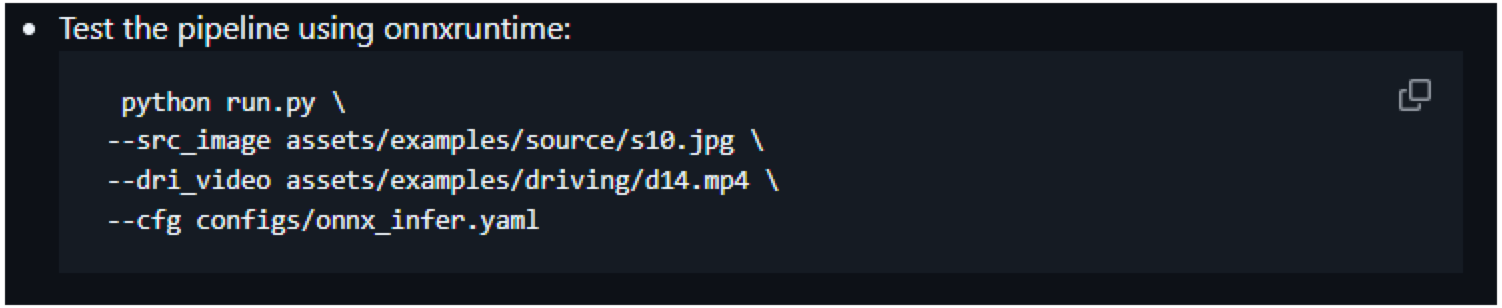


2. Onnxruntime Inference

- First, download the converted onnx model files: `huggingface-cli download warmshao/FasterLivePortrait --local-dir ./checkpoints`.
- (Ignored in Docker)If you want to use onnxruntime cpu inference, simply `pip install onnxruntime`. However, cpu inference is extremely slow and not recommended. The latest onnxruntime-gpu still doesn't support grid_sample cuda, but I found a branch that supports it. Follow these steps to install `onnxruntime-gpu` from source:
  - `git clone https://github.com/microsoft/onnxruntime`
  - `git checkout liqun/ImageDecoder-cuda`. Thanks to liqun for the grid_sample with cuda implementation!
  - Run the following commands to compile, changing `cuda_version` and `CMAKE_CUDA_ARCHITECTURES` according to your machine (your cuDNN version must be 8.x, 9.x is not compatible):

설치 잘 되었는지 확인

- python run.py --src_image assets/examples/source/s10.jpg --dri_video assets/examples/driving/d14.mp4 --cfg configs/onnx_infer.yaml



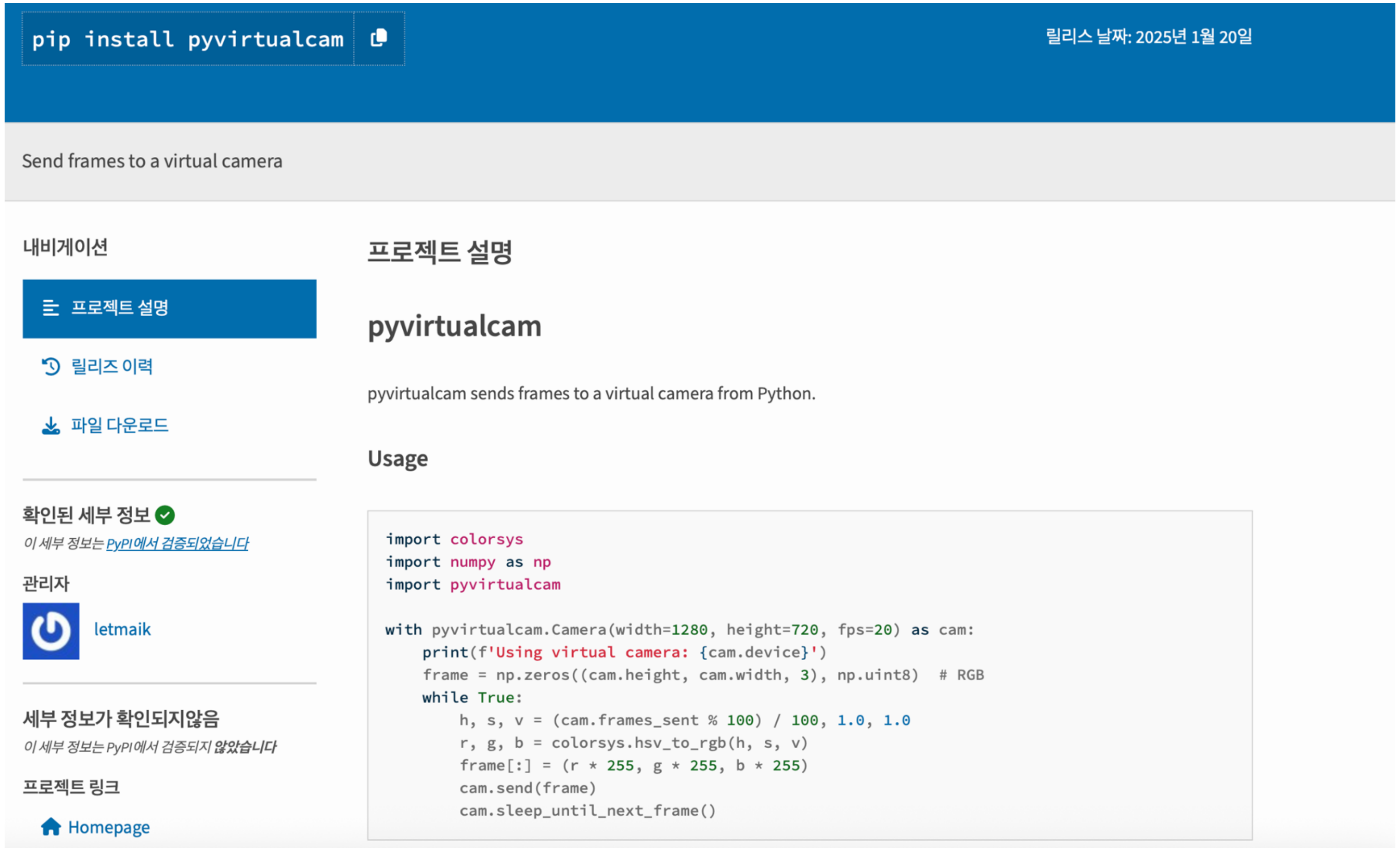
- Test the pipeline using onnxruntime:

```
 python run.py \
--src_image assets/examples/source/s10.jpg \
--dri_video assets/examples/driving/d14.mp4 \
--cfg configs/onnx_infer.yaml
```

실습해보기

1. FasterLivePortrait 구동해보기
2. FasterLivePortrait 구동 결과를 가상 카메라에 출력해보기

참고사항 가상카메라 적용 with python
pyvirtualcam



```
pip install pyvirtualcam
```

릴리스 날짜: 2025년 1월 20일

Send frames to a virtual camera

확인된 세부 정보 ✓

이 세부 정보는 PyPI에서 검증되었습니다

관리자

letmaik

세부 정보가 확인되지않음

*이 세부 정보는 PyPI에서 검증되지 **않았습니다***

프로젝트 링크

Homepage

프로젝트 설명

pyvirtualcam

pyvirtualcam sends frames to a virtual camera from Python.

Usage

```
import colorsys
import numpy as np
import pyvirtualcam

with pyvirtualcam.Camera(width=1280, height=720, fps=20) as cam:
    print(f'Using virtual camera: {cam.device}')
    frame = np.zeros((cam.height, cam.width, 3), np.uint8)  # RGB
    while True:
        h, s, v = (cam.frames_sent % 100) / 100, 1.0, 1.0
        r, g, b = colorsys.hsv_to_rgb(h, s, v)
        frame[:] = (r * 255, g * 255, b * 255)
        cam.send(frame)
        cam.sleep_until_next_frame()
```