

# Machine Learning Engineer Assignment

You are to build a **modular pipeline** that ingests a short video of a user, extracts frames and audio, applies a face reenactment model and a voice cloning model, synchronizes the generated outputs, and exposes them via an API. You will also demonstrate model optimization (e.g., export or quantization) and basic monitoring.

**Time estimate:** 8–12 hours.

**Deliver format:** GitHub repo, Jupyter notebook (for model training/testing) with working video, and a short report (~1 page).

## 1. Video & Audio Preprocessing

- Write a Python script or notebook that:
  - Accepts a short video file as input.
  - Extracts frames (OpenCV, FFMPEG) and reference audio (WAV, 16kHz).
  - Detects faces in every frame (Dlib or MediaPipe) and applies automatic face alignment.
  - Uses Whisper ASR (open-source) to transcribe the audio to text.

**Submission:** Committed code and example outputs (frames, audio, transcript).

## 2. Face Reenactment Model Integration

- Set up and run inference with an open-source face reenactment model (LivePortrait, FSRT, or DiffusionAct recommended).
  - Use extracted frames as source/driving inputs.
  - Generate an animated video segment using a target pose or random expression sequence.

**Tasks:**

- Describe any model optimization (ONNX export, TensorRT, or quantization) attempted, and record inference time and resource usage.

**Submission:** Generated animation, code, and brief notes on optimization.

## 3. Voice Cloning Model Integration

- Deploy and run inference on an open-source multilingual voice cloning model (OpenVoice, Real-Time Voice Cloning, Fish Speech, or XTTS-v2).
  - Use the reference audio to clone the speaker's voice.
  - Convert a sample transcript (from the ASR step or a provided sentence) to a new audio segment.

**Tasks:**

- Show an example of emotional or style control if supported by your chosen model.
- Attempt model export or quantization if possible, and compare inference speed versus baseline.

**Submission:** Generated speech audio and code.

## 4. API and Service Orchestration

- Wrap your pipeline into a FastAPI service with REST endpoints:
  - /upload: Accepts a video file and returns a job ID.
  - /status/{job\_id}: Returns pipeline progress/status.
  - /result/{job\_id}: Returns URLs to the processed video and audio, and the ASR transcript.
- The backend should process one request at a time (no need for async/queue/CDN but modular structure for upscaling is a plus).

**Submission:** API code (FastAPI) and example `curl` or Python requests to demonstrate the workflow.

## 5. Monitoring, Logging, and Reporting

- Integrate basic logging (Python logging or loguru).
- Add simple metrics logging: inference time for models, processing time per stage.
- Optionally, provide a dashboard (Prometheus/Grafana or a fastapi-metrics route) to visualize metrics.

**Submission:** Logs/metrics, dashboard (if implemented), and final pipeline report (describing choices, bottlenecks, and improvement areas).

## **Deliverables Summary**

Component	Required
Python code for preprocessing	<input checked="" type="checkbox"/>
Integration + test with face model	<input checked="" type="checkbox"/>
Integration + test with voice model	<input checked="" type="checkbox"/>
REST API (FastAPI) wrapper	<input checked="" type="checkbox"/>
Logs/metrics/report	<input checked="" type="checkbox"/>
Jupyter notebook (model demos/report)	<input checked="" type="checkbox"/>

## **Submission**

- Link to your public GitHub repository.
- Include environment setup instructions (requirements.txt, conda environment, etc.).
- Attach your short PDF report with Assignment working Video summarizing technical decisions, model experience, challenges, and optimization attempts.

### **Evaluation Criteria:**

- Correctness, robustness, and modularity of code.
- Demonstrated familiarity with each technology in the stack.
- Efficiency (resource usage, speed) and quality of the final outputs.
- Documentation quality and clarity of your technical report.

**Good luck! We're excited to see your applied engineering and research skills in action.**