

Documentation

Team Members:

- Sucheth Gerendla: SE21UARI189
 - Bharath Uppera: SE21UARI177
 - Tarun Arava: SE21UARI168
 - Sanjeeva Reddy: SE21UARI201
-

Introduction:

This code implements a Human Action Recognition workflow using Gradio and NVIDIA's Vision Language Model (VLM). It allows users to upload two videos, specify an action (e.g., "jumping" or "running"), and compare the recognition success rates of the action in both videos.

The Gradio interface provides a simple drag-and-drop feature for video uploads and displays real-time results, including recognition rates and a textual comparison summary.

To ensure consistency between videos of different lengths, the `extract_frames` function extracts frames at a fixed rate (e.g., 2 frames per second). In addition, the gradio interface also provides an option to trim them for uniform analysis.

Each frame is processed using NVIDIA's VLM API, which analyzes the frames and identifies the presence of the specified action.

The workflow calculates recognition success rates by dividing the number of detected frames by the total frames in each video and compares the two rates in a clear summary.

This implementation aligns with the project goals by enabling video uploads, trimming for consistent frame lengths, detecting specified actions in video frames, and displaying recognition rates and comparisons within the Gradio interface.

This system is particularly useful for evaluating the performance of action detection across different video types and lengths in an accessible and interactive way.

System Requirements:

Operating System:

Windows 10 or later, macOS, or Linux (Ubuntu 20.04+ recommended).

Hardware:

- Processor:

Multi-core CPU with at least 2.0 GHz speed (quad-core or better).

- Memory (RAM):

Minimum 8 GB (16 GB recommended for smoother execution).

- GPU (Optional):

A dedicated NVIDIA GPU for video processing (CUDA-capable recommended for efficiency).

- Storage:

At least 2 GB of free space for project files and libraries.

- Network:

Stable internet connection for API requests.

Software Requirements:

- Python Version:

Minimum Python 3.8

Recommended Python 3.10 or higher for compatibility with the latest libraries.

- Additional Tools:

FFmpeg (optional, for video-related processing tasks in case cv2 encounters codec issues).

Python Libraries Required:

- Gradio:

For building the web-based interface.

- Requests:

For making API calls to the NVIDIA endpoint.

- Opencv-python:

For video frame extraction and processing.

- Numpy:

For efficient numerical operations.

- Pillow:

For image processing and handling.

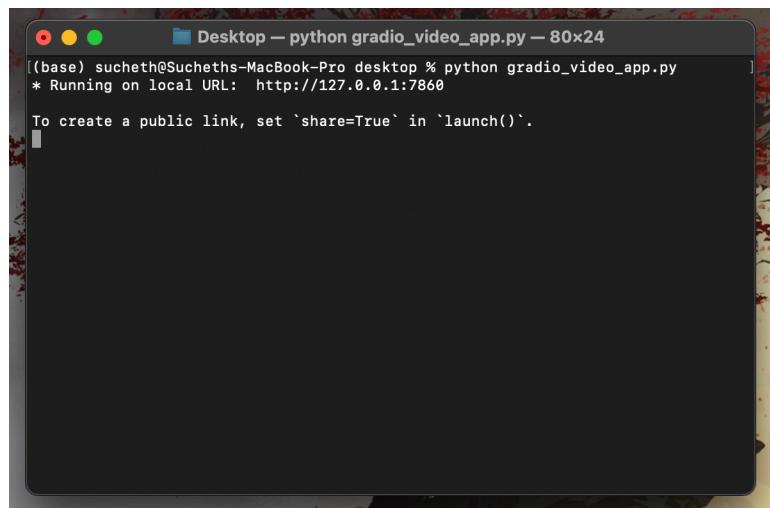
Run the following commands to install the required libraries:

```
pip install gradio requests opencv-python numpy pillow
```

How to Use:

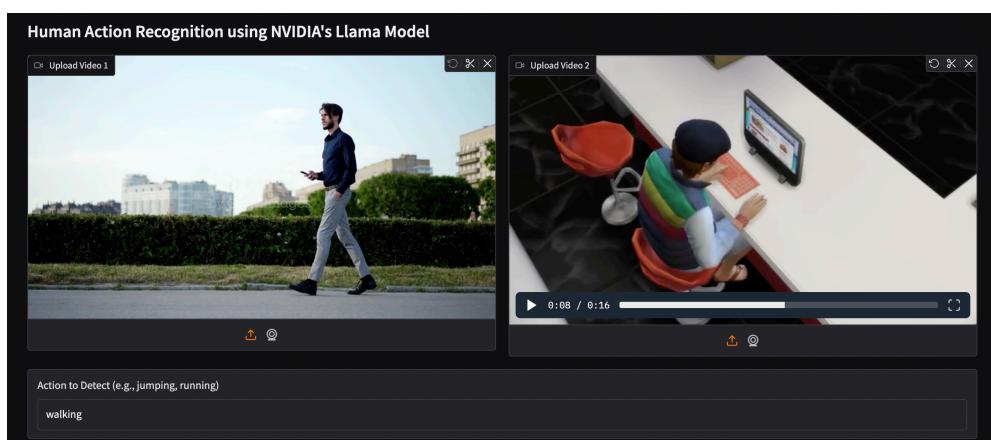
Firstly our code requires two video inputs: preferably a synthetic and a real one. Next, an action would be required to be detected in both videos. Both of these inputs, when given, the gradio interface then returns the rate of recognition of that action in those videos. Then a comparison summary is given based on the results. These recognition rates can help us understand whether that action is present in those videos or not.

To execute this code, given a MacBook's terminal, we must first navigate to the directory in which the code is present using the cd command. After that, we must run the gradio code using the command "python (code file name)". In this case, "python gradio.py".



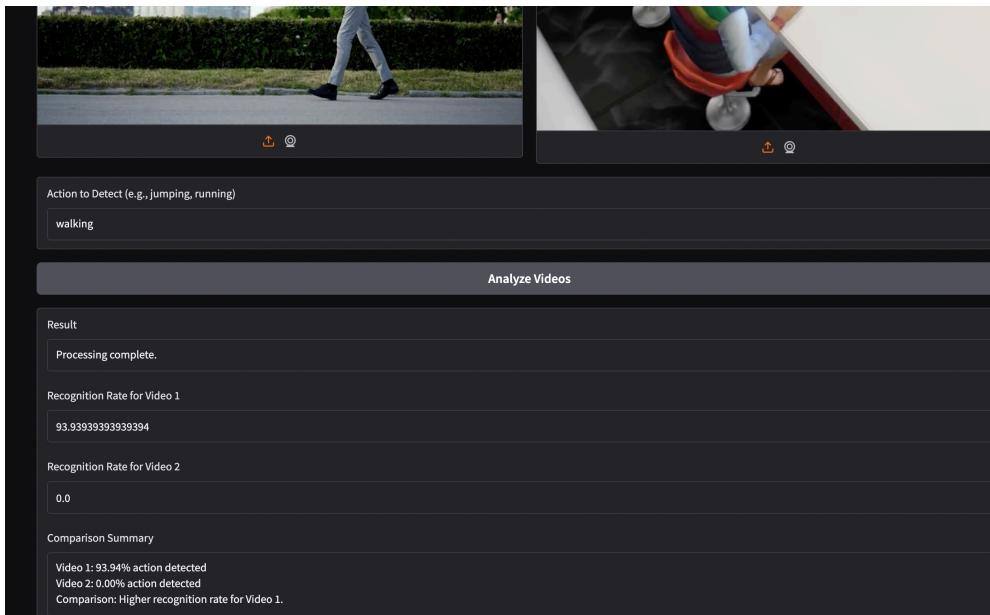
This command would give us a link which represents our gradio interface that is currently up and running. Copy and paste that link into a browser to access the interface.

With the interface opened, you can now upload your synthetic and real video. Note that there is also an option to trim these videos. Once you have uploaded the video, hover over it to see a scissors icon. This gives you the ability to trim the videos.



Next, input the action that is meant to be detected in the space provided. Click “Analyze Videos”. After a while, the interface would return the recognition rates and a comparison summary below.

In case of any errors, such as the expiration of the API key, the cause and code will be displayed in the terminal. Kindly take a look at that for further diagnosis.



Demo Video:

Youtube URL: <https://www.youtube.com/watch?v=gprojoacTiU>