**Video Comparison for Action Detection Using NVIDIA VEMO Workflow**

**Overview**

This project leverages NVIDIA VEMO Workflow to compare synthetic and real human action videos. The application trims input videos, detects specified actions, and provides a recognition success rate comparison. A user-friendly Gradio interface facilitates the process.

**Setup Guide**

**System Requirements**

1. **Python Environment**: Ensure Python 3.8 or later is installed.
2. **Dependencies**: Install the necessary Python packages by running
3. **API Key**:
   * Acquire the NVIDIA VEMO API key through official channels.
4. **Datasets**:
   * Download **Charades** and **Sims4Action** datasets as specified in the assignment.

**File Setup**

1. Save the provided Python code in a file named video\_comparator.py.
2. Place the video datasets in accessible directories for testing.

**Running the Application**

1. Run the script:
2. python video\_comparator.py
3. Open the generated URL in your browser (e.g., http://127.0.0.1:7860).

**Usage Instructions**

**Step 1: Upload Videos**

* Upload two videos using the drag-and-drop feature or file selector:
  + **Video 1**: Synthetic video.
  + **Video 2**: Real video.
* Videos must be in .mp4 format.

**Step 2: Specify an Action**

* Input the action you want to detect in the textbox (e.g., "jumping" or "running").

**Step 3: Set Trimming Parameters**

* Use the **Start Time** and **End Time** fields to specify the duration for analysis.
* Example: To analyze a clip from 10 seconds to 20 seconds, enter:
  + **Start Time**: 10
  + **End Time**: 20

**Step 4: Run the Comparison**

* Click **Compare Videos**.
* The app processes both videos and detects the action using the NVIDIA VEMO API.
* Recognition success rates are displayed for both videos.

**Expected Deliverables**

1. **Recognition Success Rates**:
   * Example:
     + **Video 1 (Synthetic)**: 85.67%
     + **Video 2 (Real)**: 78.34%
2. **Real-Time Comparison**:
   * Displayed in the interface.
3. **GIF Pairs**:
   * Generate 5 pairs of GIFs for real and synthetic video comparisons.
   * Include these in the repository for demonstration purposes.
4. **Demo Video**:
   * Create a 2-3 minute video demonstrating:
     + Uploading videos.
     + Setting parameters.
     + Running the comparison.
     + Interpreting outputs.
   * Upload the video as **unlisted** on YouTube.

**Code Explanation**

**Core Functionalities:**

1. **Video Trimming**:
   * Uses moviepy to extract the specified segment of the video.
2. **Action Detection**:
   * Sends video files and action data to the NVIDIA VEMO API.
   * Processes API responses to determine confidence scores.
3. **Interactive Interface**:
   * Built with Gradio for ease of use, supporting video input, parameter selection, and result display.

**Repository Checklist**

Ensure your GitHub repository contains:

1. **Source Code**:
   * Include video\_comparator.py with comments explaining the logic.
2. **Documentation**:
   * A README.md file covering:
     + Setup instructions.
     + Usage guide.
     + Expected outputs.
3. **Demo Video**:
   * Link to the unlisted YouTube video.
4. **Examples**:
   * Include:
     + 5 pairs of GIFs (real and synthetic comparisons).
     + Example trimmed video files.
   * Clearly label files and folders for accessibility.
5. **Access**:
   * Provide repository access to mrajendran@nvidia.com.

**Important Notes**

* Keep the demo video **unlisted** on YouTube to comply with project requirements.
* Use only the provided datasets (Charades and Sims4Action) for comparisons.