Automatic Video Montage Guided by Textual Prompts

1 Introduction

The **Automatic Video Montage Guided by Textual Prompts** project automates the creative process of video editing by leveraging advancements in **multimodal learning**, connecting textual and visual (and optionally audio) representations. This system reduces manual labor and introduces intelligent, semantic understanding of both user prompts and video content.

The core of the system is based on **cross-modal retrieval** and **similarity-based selection**, using state-of-the-art machine learning models for encoding, matching, and composing videos.

2 Technical Workflow

The pipeline consists of several main components:

2.1 Clip Extraction

The system processes the input videos by:

- Using full videos if they are short.
- Dividing longer videos into fixed-size sub-clips (e.g., 5 seconds) for uniformity.

Libraries used include `moviepy` for video operations and `opencv-python` for frame extraction if needed.

2.2 Content Representation (Encoding)

Each clip is encoded into a high-dimensional feature vector representing its semantic content. Two streams of information can be processed:

- **Visual Features**: Key frames or averaged features are encoded using a vision-language model (e.g., CLIP).

- **Audio Features** (optional): Audio is transcribed using speech-to-text models like `whisper`, and the transcript is encoded into a vector.

Models used:

- CLIP (Contrastive Language-Image Pretraining) for image and text embeddings.
- Whisper for speech-to-text transcription.

2.3 Prompt Encoding

The user-provided textual prompt is encoded using the same text encoder (e.g., CLIP's text encoder) to ensure that both prompt and clip embeddings lie within the same latent space.

2.4 Similarity Computation

Similarity between the prompt and each clip is computed using **cosine similarity**, defined as:

$$\text{Similarity}(x, y) = \frac{x \cdot y}{\|x\|\|y\|}$$

where x is the prompt embedding and y is the clip embedding.

2.5 Clip Selection

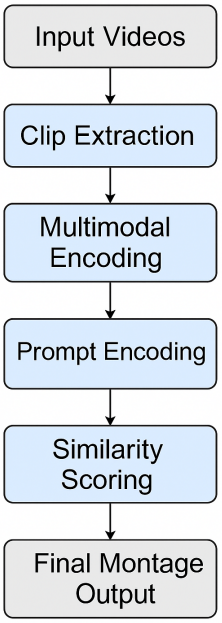
Clips are ranked based on similarity scores. The top N clips are selected according to user-specified parameters, optionally filtering out low-relevance clips through thresholding.

2.6 Montage Construction

Selected clips are sequentially assembled into a single video timeline using `moviepy`. The montage can feature simple hard cuts, with potential extensions to transitions such as crossfades.

3 Architectural Summary

The overall architecture can be summarized as:



- **Input Videos** → **Clip Extraction**
- → **Multimodal Encoding**
- → **Prompt Encoding**
- → **Similarity Scoring**
- → **Top-k Clip Selection**
- → **Montage Assembly**
- → **Final Video Output**

4 Academic Context

The project relates to several research areas:

- **Multimodal Machine Learning**: Integrating information across vision, audio, and text.
- **Cross-modal Retrieval**: Matching content from different modalities.
- **Representation Learning**: Creating generalizable embeddings via pre-trained models.

- **Self-supervised Learning**: Training without direct supervision using paired data (e.g., image-text).

Relevant references:

- Radford et al., 2021: *Learning Transferable Visual Models From Natural Language Supervision* (CLIP).
- Radford et al., 2022: *Robust Speech Recognition via Large-Scale Weak Supervision* (Whisper).

5 Strengths and Future Extensions

Strengths:

- Highly modular and extendable.
- Efficient prototyping of creative video montages.
- Flexibility in prompt specificity.

Potential Extensions:

- Dynamic shot selection based on prompt complexity.
- Integration of music synchronization.
- Style transfer for aesthetic cohesion.
- Multi-prompt montage construction.