Theoretical Analysis of Automatic Video Summarization using CLIP and Motion Detection

Luigi Daddario
Università Degli Studi di Milano-Bicocca
Email: luigi@luxdada.it

***Abstract*—This paper presents a novel pipeline for automatic video summarization that integrates semantic analysis using OpenAI's CLIP with traditional motion detection techniques. The system extracts salient segments from videos—particularly cooking videos—by eliminating static frames and selecting those that are both semantically relevant to user-defined prompts and dynamically active. We provide a comprehensive discussion of the underlying mathematical foundations, including contrastive learning, cosine similarity measures, and frame differencing. A detailed explanation of CLIP, including its architecture, training mechanism, and zero-shot capabilities, is also presented.**

1. Introduction

The exponential growth of video content has driven the need for efficient summarization techniques that capture both dynamic and semantic aspects of video data. Traditional methods based solely on low-level features often fail to capture the rich semantic context. Our approach leverages the multimodal capabilities of OpenAI's CLIP alongside classic motion detection methods to extract meaningful and dynamic video segments. This paper provides a detailed theoretical foundation, including mathematical formulations and an in-depth explanation of how CLIP works.

2. Background and Related Work

Early video summarization techniques focused on shot boundary detection and low-level feature clustering [2]. More recent methods incorporate deep learning to capture semantic information [3]. OpenAI's CLIP [1] has revolutionized multimodal learning by mapping images and text to a shared embedding space using contrastive learning. In parallel, motion detection based on frame differencing has remained a staple in computer vision for filtering non-informative content.

3. Theoretical Foundations

3.1. Contrastive Learning and CLIP

CLIP (Contrastive Language–Image Pretraining) employs a contrastive loss function to align visual and textual embeddings. Given an image x and a text prompt y, the model learns representations $f(x)$ and $g(y)$ such that matching pairs are drawn closer, while non-matching pairs are pushed apart. The training objective is expressed as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\frac{\text{sim}(f(x_i),g(y_i))}{\tau}\right)}{\sum_{j=1}^{N}\exp\left(\frac{\text{sim}(f(x_i),g(y_j))}{\tau}\right)}, \quad (1)$$

where the cosine similarity is defined as

$$\text{sim}(x,y) = \frac{f(x)\cdot g(y)}{\|f(x)\|\,\|g(y)\|}. \quad (2)$$

Here, τ is a temperature parameter, and N is the batch size. This measure captures semantic similarity by focusing on the angular distance between the embeddings.

3.2. Motion Detection via Frame Differencing

Motion detection is performed by comparing consecutive grayscale frames. Let F_t and F_{t-1} denote frames at times t and $t-1$, respectively. The absolute difference is computed as:

$$D_t(i,j) = |F_t(i,j) - F_{t-1}(i,j)|, \quad (3)$$

and the motion magnitude M_t is given by:

$$M_t = \sum_{i,j} \mathbb{1}\{D_t(i,j) > \delta\}, \quad (4)$$

where δ is a pixel-level threshold and $\mathbb{1}\{\cdot\}$ is the indicator function. A frame is considered dynamic if $M_t > \tau_{\text{motion}}$, with τ_{motion} as the motion threshold.

3.3. Segment Extraction and Merging

Candidate segments are extracted by selecting frames that satisfy both the motion and (if enabled) semantic criteria. Each segment is defined with a duration Δt centered around a key frame. To ensure temporal continuity, segments that are closely spaced are merged:

$$\text{If } t_{start}^{(i+1)} - t_{end}^{(i)} \leq \text{gap}, \text{ then merge into } S_{\text{merged}} = \left[t_{start}^{(i)}, \max\{t_{end}^{(i)}, t_{end}^{(i+1}\right. \quad (5)$$

This strategy reduces fragmentation and produces a coherent final video summary.

4. CLIP Architecture and Mechanism

OpenAI's CLIP model is a state-of-the-art multimodal framework that learns to associate images and text by aligning them in a shared embedding space. The detailed explanation from [4] can be summarized as follows:

4.1. Dual Encoder Architecture

CLIP comprises two main components:

1) **Image Encoder:** This module processes the input image using either a Convolutional Neural Network (CNN) or a Vision Transformer (ViT), extracting high-level visual features. The output is a feature vector representing the image.
2) **Text Encoder:** This module utilizes a Transformer architecture to process tokenized text, producing a feature vector that captures the semantic meaning of the text.

Both encoders transform their inputs into a common embedding space. The embeddings are then normalized, ensuring that the cosine similarity measure (which is invariant to vector magnitude) is effective in comparing the semantic content of images and text.

4.2. Contrastive Training

During training, CLIP is fed a large dataset of over 400 million (image, text) pairs. The model employs a contrastive loss, as described in Equation (1), which encourages the embeddings of matching pairs to be similar, while ensuring that non-matching pairs are dissimilar. This approach enables the model to generalize to unseen data, supporting zero-shot tasks such as image classification and semantic video summarization.

4.3. Zero-Shot Inference and Applications

One of the most powerful aspects of CLIP is its ability to perform zero-shot inference. During testing, CLIP can compare an image against a set of candidate text descriptions and determine the best match based solely on cosine similarity. This capability allows CLIP to be used for various downstream applications without additional fine-tuning.

4.4. Figures from the Medium Article

For a visual understanding of CLIP, include the following images from the Medium article by Paluchasz:

- **Figure 1: CLIP Architecture Diagram**
- **Figure 2: CLIP Flow Diagram**

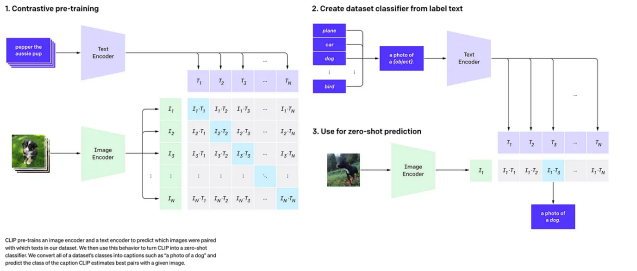

Figure 1. CLIP Architecture Diagram. This image illustrates the dual encoder structure: the image encoder (CNN or ViT) and the text encoder (Transformer), which transform inputs into a shared embedding space. Refer to Figure 1 in [4].

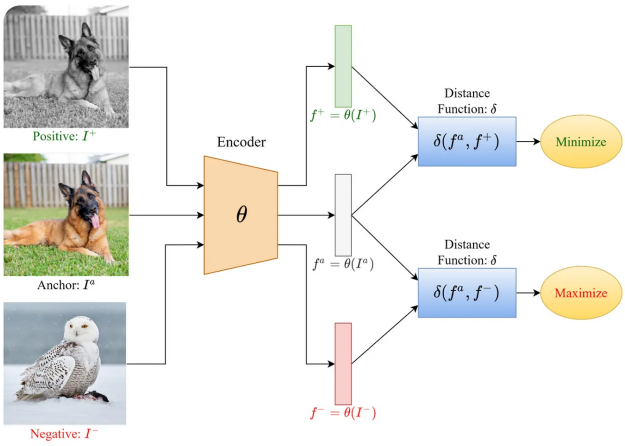

Figure 2. CLIP Flow Diagram. This image shows the contrastive training process, where the embeddings of image-text pairs are aligned via a contrastive loss function. Refer to Figure 2 in [4].

5. Methodology

Our system integrates the aforementioned components into a unified pipeline for video summarization:

1) **Frame Sampling and Preprocessing:** The video is sampled at a fixed rate. Each frame is converted to grayscale for motion detection and preprocessed for CLIP if semantic analysis is enabled.
2) **Dual Filtering:** For each frame, we compute the motion magnitude M_t and, if a text prompt is provided, the cosine similarity $\text{sim}(x, y)$ via CLIP. A frame is selected as a candidate if $M_t > \tau_{\text{motion}}$ and (when applicable) $\text{sim}(x, y) > \tau_{\text{clip}}$.
3) **Segment Extraction and Merging:** Candidate frames are expanded into segments of duration Δt and merged if temporally close.
4) **Final Assembly:** The selected segments are concatenated to form the final video summary, with a duration limit imposed if necessary.

6. Implementation Details

The pipeline is implemented in Python using:

- **OpenCV** for frame extraction and motion detection.
- **CLIP** (via PyTorch) for encoding images and text.
- **MoviePy** for video segmentation and assembly.

Videos are sourced from Google Drive, processed frame-by-frame, and the final montage is generated by concatenating the selected segments.

7. Experimental Analysis

Preliminary experiments on cooking videos indicate that:

- CLIP-based semantic filtering effectively identifies contextually significant frames.
- Motion detection removes redundant static frames, ensuring the dynamic portions of the video are highlighted.
- The segment merging strategy maintains temporal continuity, producing a coherent and concise final summary.

Future work will include quantitative evaluations and the integration of audio analysis using models such as Whisper.

8. Conclusion

We have presented a detailed theoretical and practical framework for automatic video summarization that combines CLIP-based semantic analysis with classical motion detection techniques. This paper has explored the mathematical foundations of contrastive learning, cosine similarity, and frame differencing, and provided an in-depth explanation of the inner workings of CLIP, including its dual-encoder architecture, contrastive training mechanism, and zero-shot capabilities. Future research will further refine these methods by incorporating additional modalities and advanced segmentation strategies.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[2] B. T. Truong and S. Venkatesh, "Video Summarization: A Survey," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 3, no. 1, pp. 2–19, 2007.

[3] B. Mahasseni, M. Lam, and M. S. Saquib, "Unsupervised Video Summarization with Adversarial LSTM Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] P. Paluchasz, "Understanding OpenAI's CLIP Model," *Medium*, [Online]. Available: https://medium.com/@paluchasz/understanding-openais-clip-model-6b52bade3fa3. [Accessed: Date].