***attention\_curves.py***

**Code Purpose**

This script calculates motion intensity and color contrast across video files to determine which frames a human would likely attend to most. It processes all .mp4 videos in the all\_videos directory and generates quantitative measurements for each frame:

1. **Optical Flow Calculation**: Computes frame-to-frame motion intensity using the Farneback optical flow algorithm.
2. **Color Contrast Analysis**: Creates color histograms and determines local color variation by calculating Bhattacharyya distances.
3. **Data Export**: Saves motion magnitude and color contrast values per frame per as individual JSON files per video. Optionally generates curve visualizations for both metrics.
4. **Parallel Processing**: Uses ThreadPoolExecutor from the concurrent library to process multiple videos simultaneously, significantly reducing total computation time.

This program identifies the values that will be used to extract key frames and is useful for video understanding tasks such as action recognition and scene change detection.

***KFE\_fuse.py***

**Code Purpose**

This script uses the json files created with *attention\_curves.py* and fuses the values into one attention curve per video. Then, using either the segment method or the peak method, key frames are identified from this fused curve and saved. Lastly, using the VideoMAE model from HuggingFace Transformers, each video is processed, and embeddings are extracted and saved:

1. **Optical Flow Calculation**: Computes frame-to-frame motion intensity using the Farneback optical flow algorithm.
2. **Color Contrast Analysis**: Creates color histograms and determines local color variation by calculating Bhattacharyya distances.
3. **Data Export**: Saves motion magnitude and color contrast values per frame per as individual JSON files per video. Optionally generates curve visualizations for both metrics.
4. **Parallel Processing**: Uses ThreadPoolExecutor from the concurrent library to process multiple videos simultaneously, significantly reducing total computation time.

This program identifies the values that will be used to extract key frames and is useful for video understanding tasks such as action recognition and scene change detection.

***LDA\_model.py***

**Code Purpose**

This script calculates motion intensity and color contrast across video files to determine which frames a human would likely attend to most. It processes all .mp4 videos in the all\_videos directory and generates quantitative measurements for each frame:

1. **Optical Flow Calculation**: Computes frame-to-frame motion intensity using the Farneback optical flow algorithm.
2. **Color Contrast Analysis**: Creates color histograms and determines local color variation by calculating Bhattacharyya distances.
3. **Data Export**: Saves motion magnitude and color contrast values per frame per as individual JSON files per video. Optionally generates curve visualizations for both metrics.
4. **Parallel Processing**: Uses ThreadPoolExecutor from the concurrent library to process multiple videos simultaneously, significantly reducing total computation time.

This program identifies the values that will be used to extract key frames and is useful for video understanding tasks such as action recognition and scene change detection.