Scene Detection Using OpenCV and Python

Introduction

Overview: Scene detection plays a pivotal role in video content analysis, making it easier to navigate large video files and find specific scenes quickly.

Purpose: With the growth of video content in various fields, this project aims to automate scene detection, reducing manual effort and improving efficiency.

Scope: Applications range from home videos and academic research to surveillance footage and streaming platforms.

Motivation

- **Problem Statement**: Analysing and managing extensive video content is a daunting task. Identifying specific scenes manually can be time-consuming and error-prone.
- **Objective**: To simplify the process of detecting scene transitions and creating an organized structure for video content.
- **Use Cases**:
  - **Home Videos**: Splitting long videos into individual scenes for easier navigation.

Motivation

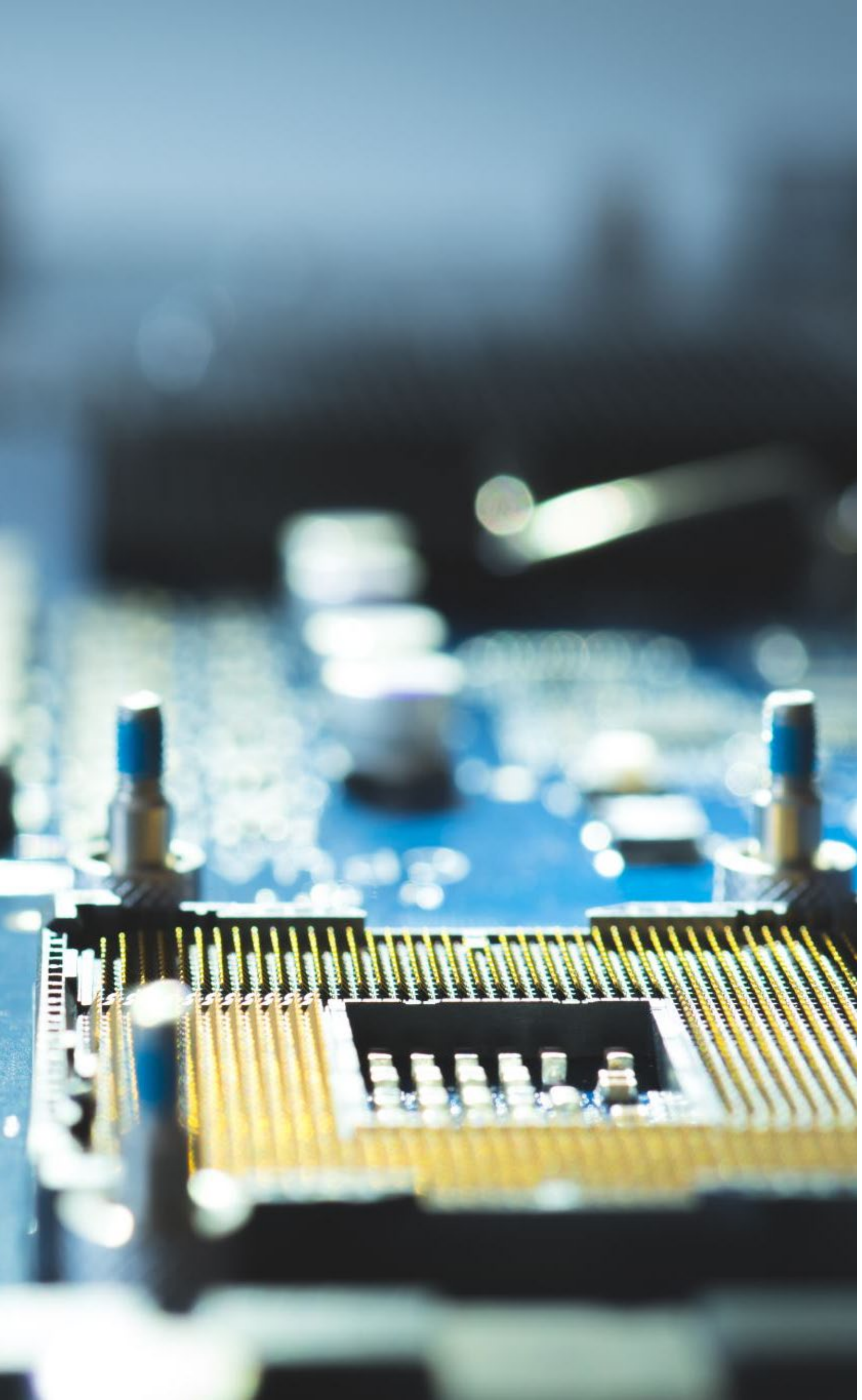
- **Commercial Removal**: Automatically detecting and removing commercials from TV recordings.
- **Surveillance Footage**: Processing and splitting footage to isolate key events.
- **Academic Analysis**: Finding mean shot lengths and analysing film structure for research.

Methodology

- **Flask Framework**: Providing a simple web interface for video uploads.
- **OpenCV Algorithms**: Utilizing OpenCV to analyze video frames and detect transitions.
- **Adaptive Detector**: An adaptive approach to find significant changes in video content.
- **Scene Detection Steps**:
  - **Video Upload**: Users upload videos through the Flask interface.

Methodology

- **Frame Analysis**: OpenCV analyzes the video frames, detecting scene transitions.
- **Calculate Frame Score**: Assigning scores to frames based on changes in hue, saturation, luminance, and edge detection.
- **Scene Cuts Detection**: Identifying scene transitions based on frame score changes.
- **Generate Output**: Providing scene start and end timestamps with scene cut start frame images.



Technologies Used

- **Flask**: A lightweight web framework used to build the application interface.
- **Python**: The primary programming language for backend development.
- **OpenCV**: A computer vision library used for video analysis and scene detection.
- **Docker**: A platform for packaging, deploying, and managing the application in consistent environments.
- **Werkzeug**: For secure file handling and URL management.
- **Pathlib**: A library for easier path manipulations.
- **Binascii**: Used for encoding and decoding file paths.

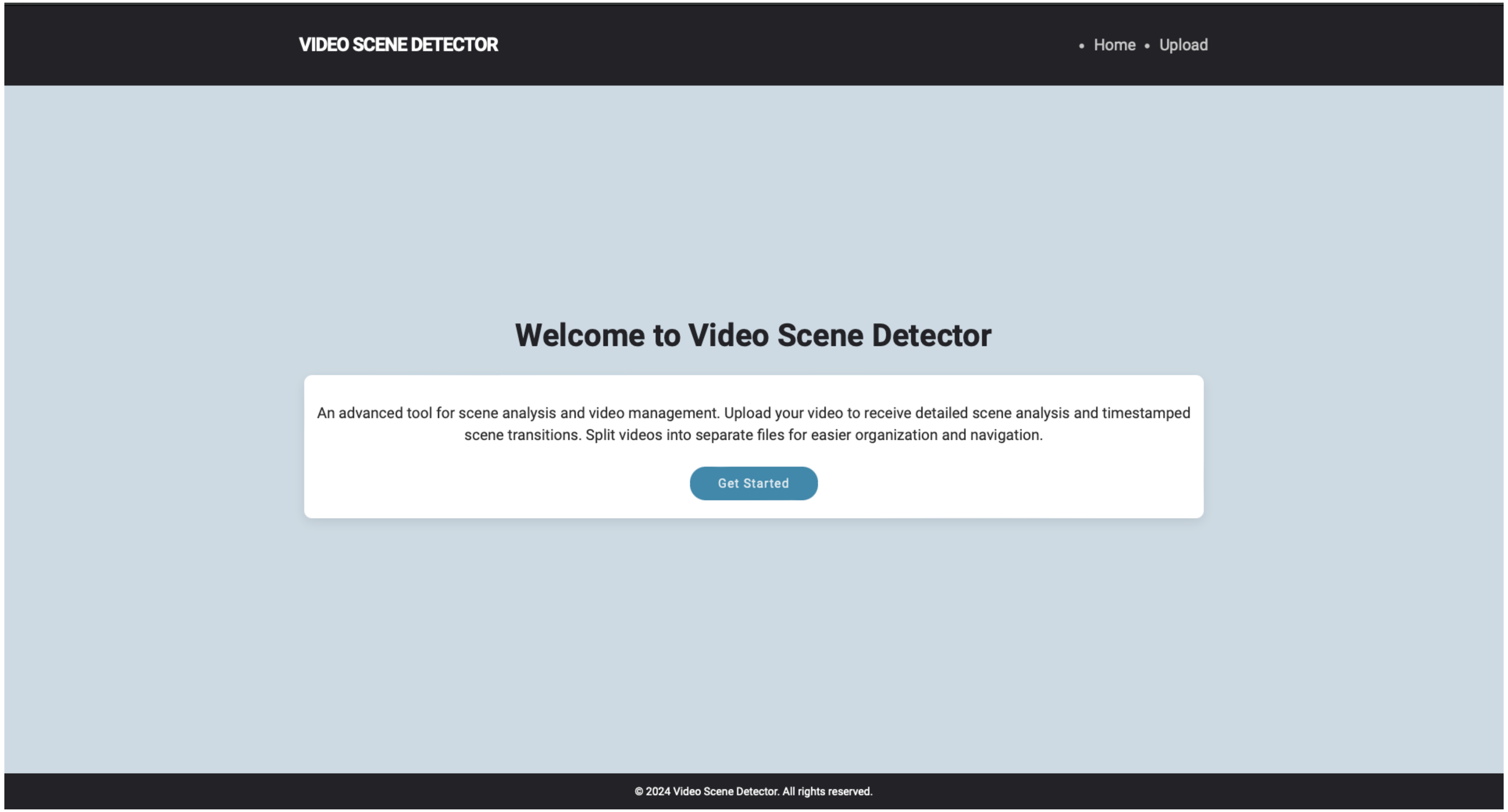
Usage

Video Upload: Users can upload video files to the application.

Parameter Configuration: Options to adjust adaptive threshold, minimum scene length, window width, and content value.

Scene Analysis: The application processes the video and provides scene transitions with timestamps and images.

Home Page

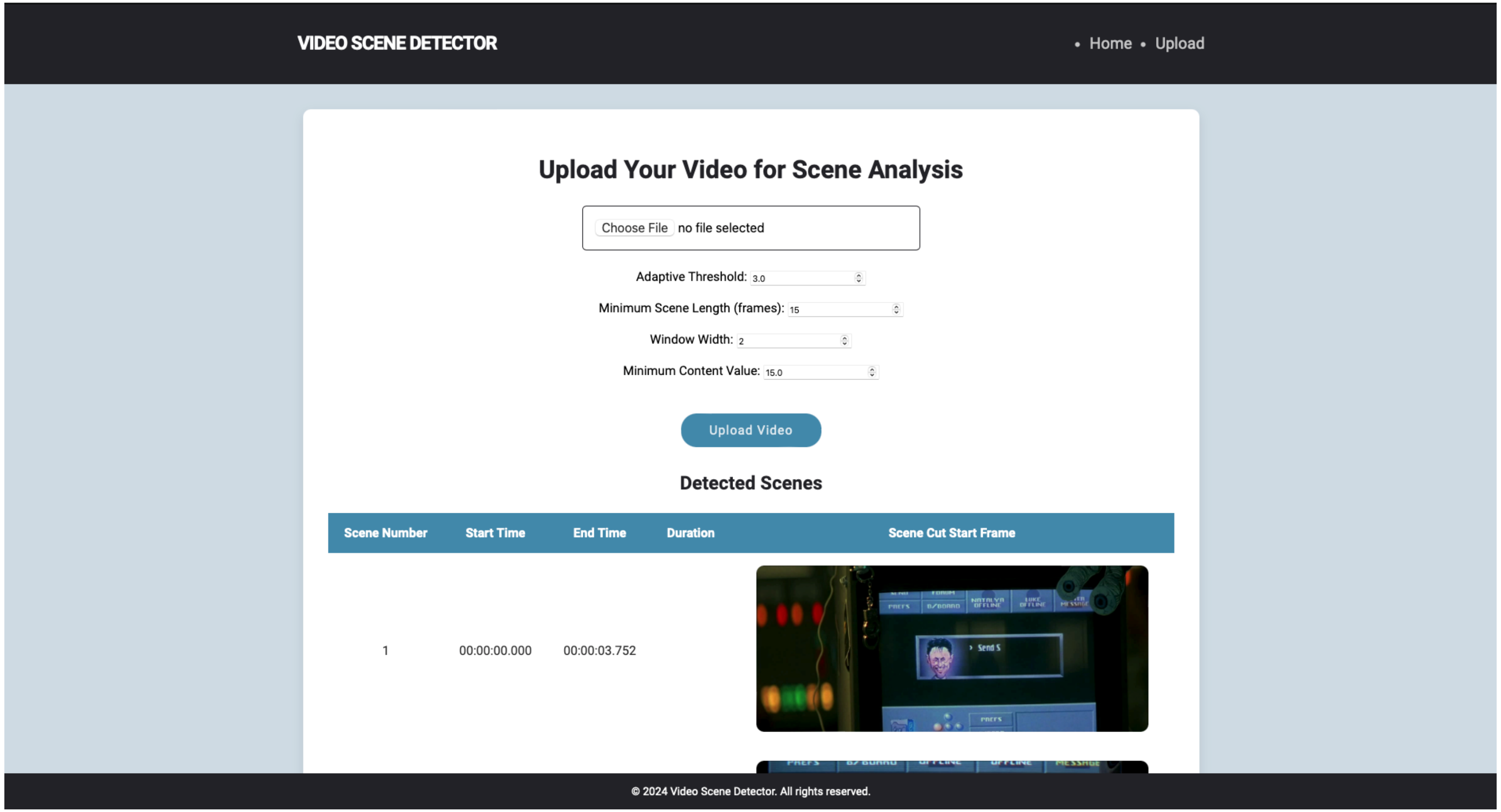


Welcome to Video Scene Detector

An advanced tool for scene analysis and video management. Upload your video to receive detailed scene analysis and timestamped scene transitions. Split videos into separate files for easier organization and navigation.

© 2024 Video Scene Detector. All rights reserved.

Upload Page



Upload Your Video for Scene Analysis

Choose File no file selected

Adaptive Threshold: 3.0

Minimum Scene Length (frames): 15

Window Width: 2

Minimum Content Value: 15.0

Detected Scenes

Scene Number	Start Time	End Time	Duration	Scene Cut Start Frame
1	00:00:00.000	00:00:03.752		

Conclusion

Benefits: This project simplifies video management and scene detection, offering various applications for personal, academic, and professional use.

Future Improvements: Opportunities to expand the feature set and integrate with other video processing tools for broader applications.

Thank You