

Guilherme de Azevedo Pereira Marques
1921160

Ferramenta de Segmentação de Vídeos por Clusters

Orientador: Sérgio Colcher

Rio de Janeiro
2021

Sumário

Especificação do programa	2
Objetivo	2
Requisitos Funcionais	2
Ler vídeos	2
Visualizar vídeos	2
Extrair características de vídeos	2
Segmentar vídeos de forma semi-automática	2
Segmentar vídeos automaticamente	3
Visualização dos clusters	3
Visualização dos clusters na linha do tempo do vídeo	3
Requisitos Não-Funcionais	3
Projeto do Programa	4
Arquitetura	4
Módulo Presentation	4
Módulo DataProcessing	5
Módulo Media	5
Diagrama de classes	6
Qualidade de Software	6
Estratégias de testes	6
Testes	7
Teste 01: Extração de Features	7
Teste 02: Clusterização	7
Teste 03: Features	7
Teste 04: Video	7
Teste 05: Npy	8
Teste 06: Utils	8

1. Especificação do programa

Nesta seção são descritos o objetivo e requisitos funcionais e não-funcionais do sistema.

1.1. Objetivo

O objetivo deste sistema é validar uma ferramenta que auxilie humanos a segmentar vídeos em cenas de forma automática, ou semi-automática a partir de clusters de segmentos de um vídeo.

1.2. Requisitos Funcionais

1.2.1. Ler vídeos

A ferramenta deverá ser capaz de ler vídeos no formato *mp4* em uma pasta especificada pelo usuário em sua máquina.

1.2.2. Visualizar vídeos

A ferramenta deverá permitir a visualização de um vídeo.

1.2.3. Extrair características de vídeos

A ferramenta deverá permitir a integração com modelos de inteligência artificial treinados para classificar vídeos, de forma que esses modelos sejam capazes de gerar vetores de características(features) que descrevem segmentos de um vídeo.

1.2.4. Segmentar vídeos de forma semi-automática

A ferramenta deverá ser capaz de, por meio de algoritmos de cluster, segmentar um vídeo a partir de um número de cenas determinado pelo usuário, de forma que cada cena seja um cluster de segmentos.

1.2.5. Segmentar vídeos automaticamente

A ferramenta deverá ser capaz de, por meio de algoritmos de cluster, segmentar um vídeo automaticamente, selecionando o número ótimo de cenas para aquele vídeo, de forma que cada cena seja um cluster de segmentos.

1.2.6. Visualização dos clusters

A ferramenta deverá permitir a visualização dos clusters de segmentos dos vídeos em um gráfico de dispersão, onde cada ponto representa um segmento.

1.2.7. Visualização dos clusters na linha do tempo do vídeo

A ferramenta deverá permitir a visualização dos clusters no formato de uma linha do tempo.

1.3. Requisitos Não-Funcionais

- Usabilidade: Associada à facilidade de uso do sistema.
- Confiabilidade: Deve-se buscar baixa frequência de falhas e robustez do sistema na recuperação de falhas.
- Padrões: Conformidade com os padrões e normas a serem seguidas no desenvolvimento do sistema.

2. Projeto do Programa

2.1. Arquitetura

Esta seção descreve a arquitetura geral da ferramenta. A Figura 1 apresenta os principais módulos da arquitetura da ferramenta. Os componentes são distribuídos em três módulos: *Presentation*, *DataProcessing* e *Media*. O módulo *Presentation* contém os componentes que estão relacionados à apresentação e interação com o usuário. O módulo *DataProcessing* possui os componentes que realizam algum tipo de transformação de dados. Por último, o módulo *Media* provê os componentes que manipulam os principais meios de dados da aplicação.



Figura 1 - Módulos da arquitetura da ferramenta

2.1.1. Módulo *Presentation*

Neste módulo os componentes tem como principal finalidade prover visualizações e interações com o usuário. Seu principal componente é o **Page**, ele é responsável por inicializar

o layout e conectar os elementos gráficos com as outras funções do sistema. O componente **Plots** é responsável por criar os principais gráficos utilizados em **Page**.

2.1.2. Módulo *DataProcessing*

Neste módulo os componentes tem como principal finalidade realizar as principais transformações e processamentos de dados do sistema. O componente **Extractors** possui como principal responsabilidade prover diferentes estratégias de extração de *features* de vídeos, ou seja, transformar os dados que estão representados em vídeo em um vetor de alta dimensionalidade. Enquanto que o componente **Clusters**, igualmente importante, tem como responsabilidade prover diferentes estratégias para clusterizar dados.

2.1.3. Módulo *Media*

Neste módulo os componentes têm como principal finalidade representar abstrações de diferentes meios de dados. O componente **Video** possui a responsabilidade de interação com vídeos, nele estão presentes funcionalidades como de leitura de frames e acesso a metadados de um arquivo de vídeo. O componente **Features** possui a responsabilidade de representar as features extraídas de um vídeo, provê funcionalidades de leitura e escrita, e também de metadados relacionados à features de um vídeo específico.

2.2. Diagrama de classes

Esta seção apresenta o diagrama de classes da ferramenta. Vale ressaltar que na Figura 2 estão apenas as classes presentes nos principais módulos.

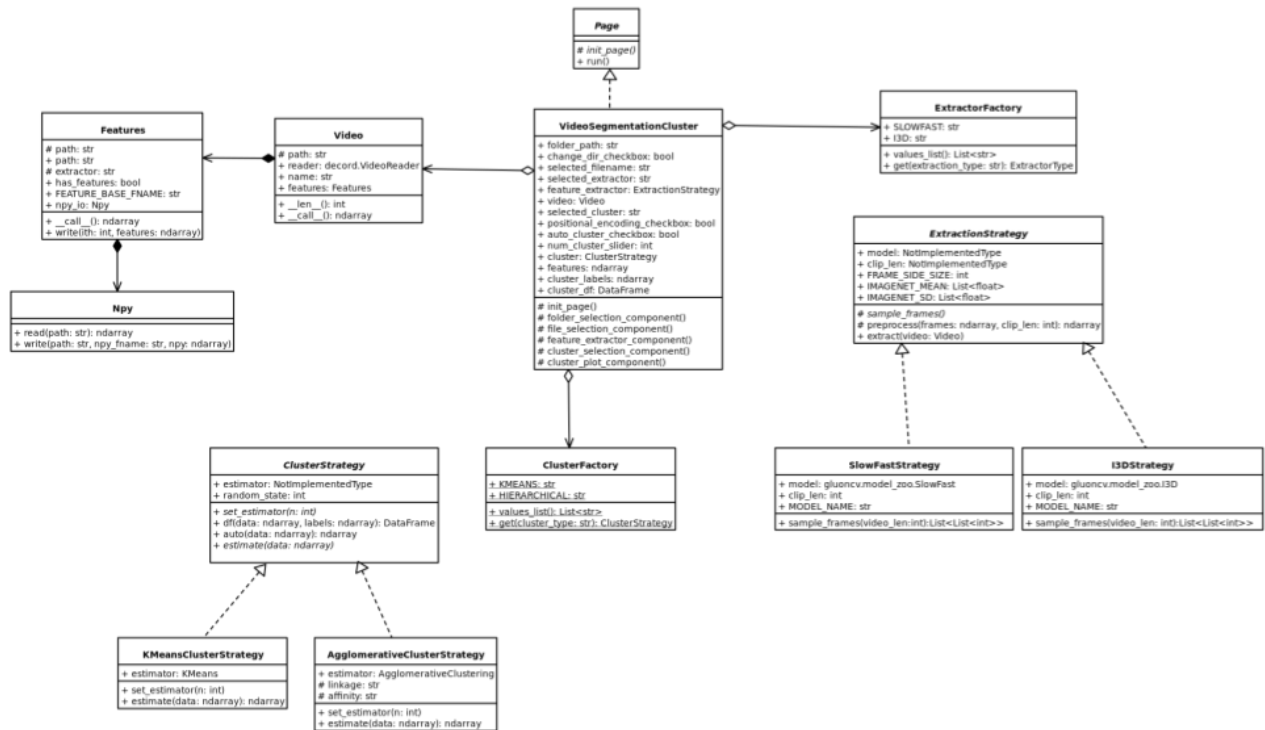


Figura 2 - Diagrama de classes da ferramenta

3. Qualidade de Software

3.1. Estratégias de testes

As estratégias de testes implementadas neste projeto foram as de unidade e integração. Para tanto foi utilizado o framework nativo de python *unittest*. A seguir, na seção 3.2, são descritos os testes realizados.

Além disso, para testes cuja automação é complexa foram utilizados testes manuais exploratórios.

3.2. Testes

Esta seção traz a descrição de cada um dos casos de testes que devem ser utilizados durante o desenvolvimento do sistema. O código fonte dos testes estão disponíveis na pasta "src/tests".

Para executá-los basta ir para a pasta **src** na raiz do projeto e executar o comando "python unittest discover tests/".

3.2.1. Teste 01: Extração de Features

Descrito no arquivo "test_extractors.py", este módulo contém um método de teste para cada uma das seguintes funcionalidades: extração de features com o modelo *SlowFast*, extração de features com o modelo *I3D* e a construção de objetos de extração de features com a classe *ExtractorFactory*.

3.2.2. Teste 02: Clusterização

Descrito no arquivo "test_clusters.py", este módulo contém um método de teste para cada uma das seguintes funcionalidades: clusterização com *KMeans* com número de clusters pré-determinado, clusterização com *AgglomerativeClustering* com número de clusters pré-determinado, clusterização com o método de determinação do número ótimo de clusters e a construção de objetos de clusterização com a classe *ClusterFactory*.

3.2.3. Teste 03: Features

Descrito no arquivo "test_features.py", este módulo contém um método de teste para cada uma das seguintes funcionalidades: o path que as features serão salvas e lidas é o path esperado, e leitura e escrita dos arquivos que guardam as features.

3.2.4. Teste 04: Video

Descrito no arquivo "test_video.py", este módulo contém um método de teste para cada uma das seguintes funcionalidades: o comprimento em frames do vídeo é o esperado e, leitura dos frames de vídeos.

3.2.5. Teste 05: Npy

Descrito no arquivo "test_npy.py", este módulo contém um método de teste para cada uma das seguintes funcionalidades: leitura de arquivos npy, escrita de arquivos npy e contagem de arquivos npy em uma pasta.

3.2.6. Teste 06: Utils

Descrito no arquivo "test_utils.py", este módulo contém um método de teste para cada uma das seguintes funcionalidades: determinação de segmentos contínuos de mesmo cluster e *positional encoding*.