Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Uma ferramenta de estilos para sumarização automática de vídeo

Gabriel Fritz Sluzala

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. Díbio Leandro Borges

Brasília
2019

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Uma ferramenta de estilos para sumarização automática de vídeo

Gabriel Fritz Sluzala

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. Dr. Díbio Leandro Borges (Orientador)
CIC/UnB

Prof. Dr. Vinícius Ruela Pereira Borges
Universidade de Brasília

Prof. Dr. Marcelo Ladeira
Universidade de Brasília

Prof. Dr. José Edil Guimarães Medeiros
Coordenador do Curso de Engenharia da Computação

Brasília, 15 de Fevereiro de 2019

Dedicatória

Dedico este trabalho ao meu filho: Julian, o qual, ansioso, espero a sua chegada.

Agradecimentos

Agradeço ao meu pai **Alceu Sluzala**. Obrigado por estar sempre comigo, principalmente nos momentos difíceis.

Agradeço à minha esposa **Vanessa Fritz Cavalcante**. Obrigado por me lembrar de valorizar as coisas certas.

Por fim, agradeço imensamente ao meu orientador, **Díbio Leandro Borges**.

Resumo

Sumarização automática de vídeos consiste em aplicar técnicas capazes de gerar versões compactas e representativas do vídeo original. Essa vem sendo explorada em diversas pesquisas atualmente, devido a necessidade de consumo de grandes quantidades de dados em formato de vídeo. Neste contexto, diversos trabalhos em aprendizagem de máquina comparam seus resultados usando bases de referência, as quais apresentam vídeos originais e sumários informativos, a fim de verificar a eficiência de seus modelos. Porém, pouco são exploradas as diferentes necessidades de sumarização, que não relacionadas com a seleção de partes mais informativas do vídeo. Neste trabalho, introduz-se o conceito de estilos aplicados a sumarização, os quais são capazes de gerar sumários para diferentes necessidades. Em seguida, utiliza-se um modelo supervisionado para gerar escores de importância para os quadros de vídeos e, com isso, desenvolvem-se critérios de seleção a fim de gerar sumários diferentes, de acordo com a finalidade desses. Por fim, realiza-se uma análise dos diferentes sumários gerados.

Palavras-chave: sumarização de vídeo; aprendizagem supervisionada; sumarização baseada em estilo; critérios de seleção

Abstract

Automatic video summarization is the application of techniques, which can create representative and compact versions of the original video. It has been significantly explored in researches, due to the need of absorbing great amounts of data in video format. In this context, several researches in machine learning compare their results to benchmarks, which presents the original videos and the informative summaries, in order to test their models. However, the different needs of summarization, other than the selection of the video's most informative parts, are little explored. It is proposed a summarization applied style concept, which is capable of creating summaries for different needs. Then, a supervised model is used to generate the importance scores of frames and, thereby, selection criterias are developed in order to create different summaries, according to its purpose. Lastly, the different summaries, generated by the selection criterias, are analysed.

Keywords: video summarization; supervised learning; style-based summarization; selection criteria

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

CNN Rede Neuronal Convolucional.

DPP Ponto de Processo Determinante.

KTS Segmentação Temporal de *Kernel*.

LSTM Long Short-Term Memory.

MLP Multi Layer Perceptron.

RNA Rede Neuronal Artificial.

RNN Rede Neuronal Recorrente.

Capítulo 1

Introdução

1.1 Contextualização do problema

Vídeo é uma forma muito poderosa de transmissão de informação. Por ser muito versátil, apresenta inúmeras utilidades em diversos setores (e.g., entretenimento, educação, ciência, indústria, segurança, defesa e artes). Uma vez que maneiras de gravar e armazenar vídeos estão se tornando mais baratas, a presença deles como meio informativo apresenta uma relevância crescente [6].

Neste contexto, um problema surge quando há a necessidade de consumir grandes quantidades de vídeo. Uma vez que um vídeo têm uma demanda em tempo para assisti-lo, não é possível consumir exaustivamente muitos deles em tempo hábil. Por exemplo: Seriam necessários 82 anos para assistir todos os vídeos produzidos no Youtube [7] em um dia [5]! Além disso, o gráfico de barras da Figura 1.1 mostra o crescimento do número de artigos publicados, os quais incluem como tópico *vídeo*, ao longo dos anos.

Buscando solucionar esse problema, técnicas e ferramentas que aumentem a eficiência do consumo das informações contidas em vídeo são necessárias para lidar com o volume de dados produzidos. Uma das técnicas, muito estudada em pesquisas atuais, é a sumarização automática de vídeos.

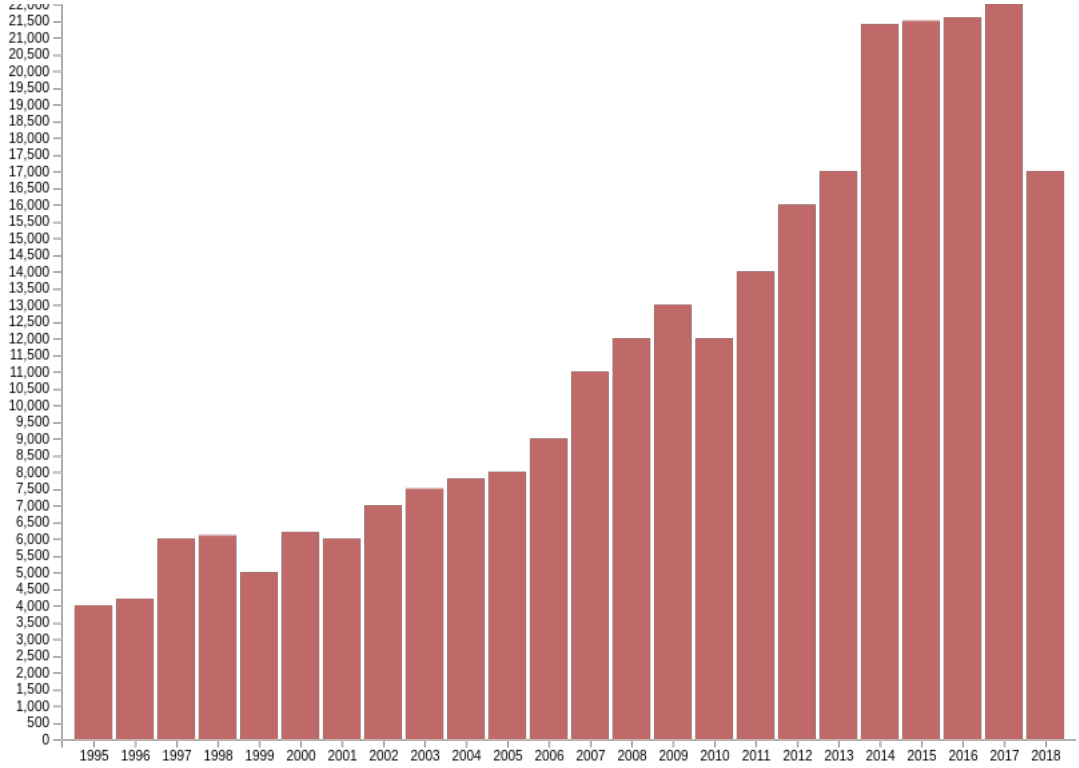

Figura 1.1: Número de artigos publicados (Eixo Y) nos últimos 25 anos (Eixo X). Os dados foram obtidos no Web of Science [1].

1.2 Definição do problema

Um sumário de um vídeo é uma versão compacta que apresenta informações do vídeo original de uma maneira visual [5]. Existem duas principais formas de se sumarizar um vídeo: sumarização baseada em quadros (imagens estáticas que compõem o vídeo) [8, 9, 10, 11, 12] e sumarização baseada em cenas (conjuntos contíguos de quadros no tempo) [13, 14, 15, 6]. Na sumarização baseada em cenas, seleciona-se os conjuntos contíguos de quadros mais importantes do vídeo, enquanto que a sumarização baseada em quadros seleciona-se os principais quadros do vídeo.

Sob a perspectiva de abordagens automáticas de sumarização, existem duas principais: Técnicas não-supervisionadas e supervisionadas [5]. Diversas abordagens usando técnicas não-supervisionadas foram abordadas em trabalhos anteriores, as quais usam critérios intuitivos para selecionar os quadros mais importantes do vídeo [8, 12, 13, 6, 16, 17, 18, 19, 20]. Dentre outras, técnicas supervisionadas estão sendo investigadas recentemente. Nelas, em contraposição às não-supervisionadas, busca-se aprender a partir de sumários criados por humanos, a fim de buscar uma representação abstrata dos dados que permita identificar um critério de seleção alinhado com o entendimento humano do conteúdo dos

vídeos [9, 21, 22, 23, 24].

Neste contexto, abordagens supervisionadas para sumarização de vídeo precisam responder três perguntas principais: *Que tipo de modelos devem ser usados?*, *Como obter dados suficientes para treinar um modelo que generalize bem?* e *Como encontrar uma representação de dados abstrata que leve em consideração a interdependência dos quadros (a fim de possibilitar uma escolha de escores de importância eficiente)?* [5].

Um dos trabalhos que abordou todos esses aspectos foi o realizado por Zhang et al. [5]. No desenvolvimento de seu modelo dppLSTM, Zhang et al. investigou como aplicar *Long Short-Term Memory* (LSTM) e suas variantes para realizar sumarização de vídeos supervisionada, demonstrando que o aspecto sequencial da LSTM, combinado com Ponto de Processo Determinante (DPP) para modelar seleção diversa de subconjuntos, é essencial para se obter resultados melhores. O modelo dppLSTM, resultante de seu trabalho, quando publicado, obteve os melhores resultados em duas bases de dados de referência. Além disso, Zhang et al. apresentou abordagens para o problema de anotações insuficientes, combinando bases de dados e usando técnicas de adaptação de domínio [5].

Em sua saída, o modelo dppLSTM gera um subconjunto de quadros-chave, os quais são convertidos em cenas-chave. As cenas-chave são, então, selecionadas segundo um critério de seleção pré-definido para gerar o sumário. Neste cenário, percebeu-se que os trabalhos de sumarização supervisionada propostos até então produzem modelos capazes de definir escores de importância para quadros/cenas e, em seguida, apresentam critérios de seleção desses, a fim de resumir o vídeo. Esses critérios são utilizados, sempre, com o intuito de obter melhores performances nas bases de referência. Algo que não foi explorado, porém, é o potencial que esses critérios têm de **produzir sumários diferentes** a partir dos escores de importância apresentados pelos modelos. Observe que o objetivo de se gerar tais sumários não seria de obter uma acurácia maior nas bases de referência, mas sim de produzir diversos vídeos compactos, a partir de um vídeo original, os quais apresentam informações diferentes, cada um com suas próprias características/peculiaridades. Neste sentido, poderia-se pensar em diferentes **estilos de sumarização**.

Para entender melhor a necessidade de se produzir sumários diferentes, imagine a seguinte situação: suponha que um *trailer* (logo, um tipo de sumário) de *O Senhor dos Anéis: O Retorno do Rei* (um filme) [25] precise ser feito. Para isso, seria necessário assistir o filme e, com isso, identificar quais cenas são as mais relevantes e quais são irrelevantes para o **entendimento** do filme. Porém, ao fazê-lo, não poderia apresentar todas as partes mais relevantes do filme, pois, dessa forma, não despertaria o interesse do público de assisti-lo (para quê assistir um filme que você já sabe tudo de importante que vai acontecer?). Neste cenário, uma das maneiras de se imaginar o problema de fazer um *trailer* a partir de um filme é: **Como sumarizar o filme, sabendo quais partes dele**

são mais importantes/não importantes para a história, de uma forma que não revele a história inteira e, ao mesmo tempo, cative a atenção do público?

Percebe-se com esse exemplo que, em se tratando de sumarização, existem diversas necessidades diferentes que vão além de apresentar as partes mais informativas. É possível pensar em **estilos diferentes** de sumarização, a partir das importâncias dos quadros/cenas. Esses estilos teriam o objetivo de atender necessidades diferentes de sumarização.

Com base no que foi apresentado nesta seção, o problema abordado neste trabalho é **a produção de sumários com estilos diferentes, a partir dos escores de importância das cenas do vídeo original**.

1.3 Objetivo do trabalho

Este trabalho tem como objetivo: *(i)* introduzir o conceito de **estilo** em sumarização de vídeos supervisionada, *(ii)* Usar o modelo de sumarização de Zhang et al. [5], o qual foi disponibilizado abertamente [26], como ponto de partida e, a partir das importâncias das cenas apresentadas na saída do modelo, criar critérios de seleção de cenas, que não o utilizado por Zhang et al., e (iii) aplicar os critérios de seleção em vídeos da base de referência SumMe [23] e verificar os sumários gerados, apresentando suas características únicas, as quais podem ser entendidas como estilos diferentes.

1.4 Apresentação do manuscrito

No Capítulo 2, são revisados os principais conceitos de *aprendizagem de máquina* fundamentais para o entendimento deste trabalho, com foco em *aprendizagem supervisionada*.

Em seguida, no Capítulo 3, aprofunda-se em uma classe de algoritmos supervisionados específica: As Redes Neuronais Artificiais (RNA's), revisando a forma com que elas aprendem a partir de dados e apresentando uma classe de RNA: As Redes Neuronais Artificiais profundas. Por fim, abordam-se duas arquiteturas de RNA's profundas: A Rede Neuronal Convolucional (CNN) e a Rede Neuronal Recorrente (RNN), devido a importância dessas arquiteturas, e dos módulos que as compõem, para a compreensão deste trabalho.

No Capítulo 4, apresentam-se os trabalhos relacionados, com um foco no trabalho realizado por Zhang et al.. O principal objetivo do capítulo é apresentar o modelo dppLSTM.

No Capítulo 5, fundamenta-se o conceito de estilo e apresenta-se a metodologia proposta para criar sumários com estilos diferentes, a partir dos escores de importância das cenas. Em seguida, no Capítulo 6, apresentam-se os principais resultados experimentais obtidos e as análises qualitativas e quantitativas.

Por fim, o Capítulo 7 apresenta as conclusões, evidenciando as contribuições do trabalho. Em seguida, discutem-se os trabalhos futuros.

Capítulo 2

Aprendizagem de Máquina

Este capítulo tem como objetivo apresentar os fundamentos de aprendizagem de máquina necessários para o entendimento da metodologia proposta neste trabalho. Primeiramente, apresenta-se a definição de aprendizagem de máquina. Com isso, descreve-se o que são *learning algorithms* (algoritmos de aprendizagem). Em seguida, explicam-se os paradigmas da aprendizagem de máquina: supervisionado e não-supervisionado. Por fim, apresenta-se mais detalhadamente o funcionamento de um algoritmo supervisionado, a regressão linear, a fim de possibilitar a melhor compreensão desta classe de algoritmos.

2.1 Definição

O estudo e a modelagem computacional de processos de aprendizagem em suas múltiplas manifestações constituem o domínio da aprendizagem de máquina, referenciada também como *machine learning*. Essa apresenta grande significância em diversas áreas que se beneficiam da teoria e da modelagem computacional de processos de aprendizagem (i.e., ciência cognitiva, inteligência artificial, ciência da informação, reconhecimento de padrões, psicologia, educação, epistemologia, filosofia) [27].

2.2 *Learning algorithms*

É dito que um programa de computador aprende da experiência (E) com respeito a uma classe de tarefas (T) e medida de performance (P), se a sua performance em T melhora com E, usando como parâmetro de melhoria a medida quantitativa P [2]. Tarefas, em aprendizagem de máquina, usualmente, são descritas em termos de como o algoritmo deve processar a experiência. De maneira geral, experiência é descrita como **exemplos** (observações constituídas por um conjunto de atributos medidos quantitativamente) que se apresentam ao algoritmo, possibilitando que este melhore sua performance em T. Por

fim, performance (P) é uma maneira quantitativa de medir o desempenho do algoritmo em T [28].

2.3 Paradigmas de aprendizagem de máquina

Algoritmos de aprendizagem de máquina podem, de maneira geral, ser categorizados em duas classes: Supervisionado e não-supervisionado. Na maioria dos algoritmos de aprendizagem, a experiência é uma **base de dados**, uma coleção de exemplos (*data points*). Essa base pode ser vista como uma matriz com M linhas e N colunas (MxN), contendo M exemplos e N atributos (características do exemplo medidas quantitativamente), denominada **matriz de modelo**. Neste cenário, as classes de algoritmos podem ser definidas como [2]:

- **Não-supervisionado:** Algoritmos que experienciam uma base de dados e, com isso, aprendem padrões úteis em sua estrutura. Uma das principais tarefas realizadas por algoritmos não-supervisionados é o agrupamento (*clustering*), o qual consiste em dividir a base de dados em agrupamentos compostos por exemplos similares.
- **Supervisionado:** Algoritmos que experienciam uma base de dados, na qual cada exemplo é previamente associado à um **valor alvo** (discreto ou contínuo). Nisso, aprendem a descobrir **o valor alvo de exemplos não experienciados previamente**.

2.4 Regressão linear

Como a aprendizagem supervisionada é a mais relevante para o entendimento deste trabalho, apresenta-se, nesta seção, um exemplo concreto de algoritmo pertencente a essa classe: A regressão linear.

Regressão linear é um método de se aprender um modelo, na forma de uma função linear, capaz de predizer o valor de $\mathbf{y} \in \mathbb{R}$ (valor alvo) correspondente a um exemplo $\mathbf{x} \in \mathbb{R}^n$. A fórmula abaixo apresenta, de maneira clara, como a função linear calcula, a partir do exemplo $\mathbf{x}$, $\hat{\mathbf{y}}$ (predição do valor de $\mathbf{y}$):

$$\hat{y} = \mathbf{w}^T x \tag{2.1}$$

Onde, $\mathbf{w} \in \mathbb{R}^n$ é o **vetor de parâmetros**, um conjunto de pesos que determina como cada atributo x_i afeta a predição. Se um atributo possui um peso de grande magnitude relacionado a ele, então ele possui um grande efeito na predição. Caso seu peso seja zero, significa que o atributo não afeta a predição. Um exemplo de tarefa que compete

a regressão linear é: Predizer o preço de uma casa baseada no seu tamanho em metros quadrados (m^2).

Para realizar essa predição corretamente (ou seja, produzir um valor de uma casa, a partir de seu tamanho, corretamente), é necessário aprender qual é o vetor de parâmetros adequado. Durante a aprendizagem, utilizam-se, como exemplos, uma base de dados de treinamento $X^{train} \in \mathbb{R}^{n \times m}$ contendo n exemplos e m atributos, e o vetor de alvos y^{train}, contendo um valor alvo para cada exemplo em X^{train}.

Com os parâmetros aprendidos, o modelo é capaz de realizar predições em exemplos que não foram experienciados durante o treinamento. Para se avaliar o modelo, usa-se uma base de dados X^{teste} e o vetor de alvos y^{teste}, contendo exemplos não usados durante o treinamento [2].

2.4.1 Medida de performance

Para se realizar a aprendizagem e, em seguida, avaliá-lo, é necessário definir medidas de performance. Uma das formas de medir performance durante o treinamento é definir uma **função de custo** J: $\mathbb{R} \rightarrow \mathbb{R}$:

$$J(\mathbf{w}) = \frac{1}{2m} \sum_i (\mathbf{w}^T x - y)_i^2 \tag{2.2}$$

Para se avaliar o modelo aprendido, comumente, usa-se o **erro quadrático médio** na base de teste:

$$MSE_{teste} = \frac{1}{m} \sum_i (\hat{y} - y)_i^2 \tag{2.3}$$

Com isso, note que o objetivo da aprendizagem é reduzir o valor de MSE_{teste}. Para reduzir esse valor, é necessário usar um algoritmo de aprendizagem capaz de encontrar um vetor de parâmetros que reduza, em suas etapas, os valores de $J(\mathbf{w})$ e, com isso, no final da aprendizagem, produza um valor de MSE_{teste} reduzido. Um algoritmo capaz de encontrar um vetor de parâmetros ótimo é o **Método da Descida de Gradiente**. A aplicação desse método produz um vetor de parâmetros **w*** que minimiza a função de custo J [2].

2.4.2 Método da Descida de gradiente

O método da descida de gradiente é um algoritmo que possibilita minimizar uma função diferenciável $f(\mathbf{x})$ a partir de seu gradiente $\nabla_x f$, o vetor de derivadas parciais de f. Para realizar a aprendizagem, o método minimiza a função de custo $J(w)$ a partir de $\nabla_w J$. No caso, o gradiente da função de custo indica em que direção deve-se ajustar os valores

de w para que o valor de J **aumente**. Como, no caso, deseja-se **diminuir** o valor de J, ajusta-se o vetor de parâmetros **w** da seguinte maneira:

$$w = w - \theta \nabla_w J, \tag{2.4}$$

em que θ é a taxa de aprendizagem, uma valor entre 0 e 1, o qual define o grau de ajuste em cada iteração. Quanto maior o valor de θ, maior é o ajuste de **w** em cada etapa da aprendizagem. Com isso, em cada iteração, os valores de **w** são definidos, de maneira que o valor da função de custo seja menor que na etapa anterior. O treinamento é interrompido quando $\nabla_w J$ **atinge um mínimo local**.

Capítulo 3

Redes Neuronais Artificiais

Rede Neuronal Artificial (RNA) é um modelo de aprendizado inspirado nos neurônios biológicos componentes do sistema nervoso central, os quais são essenciais para a aprendizagem em animais [29]. É possível, a partir deste modelo, realizar uma otimização de seus parâmetros utilizando um algoritmo de aprendizagem (e.g., Método da Descida de Gradiente), a fim de resolver uma grande variedade de tarefas não lineares (e.g., classificar imagens, sumarizar vídeos).

Até 2006, o treinamento de RNA's era muito limitado pela demanda computacional de arquiteturas mais complexas, porém, com a melhoria no processamento e o avanço em aprendizagem profunda (e.g., técnicas em inicialização de parâmetros [30]), foi possível obter performances expressivamente melhores em diversas tarefas em visão computacional, reconhecimento de discurso e processamento de linguagem natural, superando outras soluções tradicionais.

As seções a seguir introduzem os fundamentos básicos sobre RNA. A primeira seção apresenta o *perceptron*, o modelo mais simples de RNA. Em seguida, explica-se sobre neurônios sigmóides e *Multi Layer Perceptron* (MLP). Por fim, discute-se sobre aprendizagem profunda, apresentando dois tipos de RNA's profundas: Rede Neuronal Convolucional (CNN) e Rede Neuronal Recorrente (RNN).

3.1 *Perceptrons*

Publicado em 1958 por Frank Rosenblatt [31], *Perceptron* é o tipo mais simples de RNA. Um *Perceptron* recebe, em sua entrada, N valores $x_i \in \{0, 1\}$, e produz, em sua saída, $y \in \{0, 1\}$, como mostrado na Figura 3.1. Para computar a saída, o *Perceptron* associa, a cada entrada, um peso $w_i \in \mathbb{R}$. Com isso, a saída é definida como sendo 1 ou 0, de acordo com o valor da soma ponderada $z = \sum_{i=1}^{N} w_i.x_i$. Se o valor de z for maior ou igual

um certo valor de corte, a saída y é igual à 1, caso contrário, y é igual à 0. A Equação 3.1 demonstra isso em termos algébricos.

$$\mathbf{y} = \begin{cases} 1 & \text{se} \quad \sum_{i=1}^{N} w_i.x_i \geq corte, \\ 0 & \text{caso contrário} \end{cases} \tag{3.1}$$

O modelo matemático do *perceptron* pode ser pensado como um sistema capaz de tomar uma decisão binária (y) por meio do ponderamento (w_i) de algumas evidências (x_i). Variando os valores do corte e dos pesos, obtemos modelos diferentes de tomada de decisão.

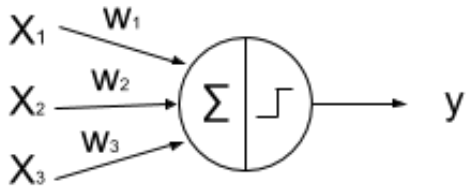

Figura 3.1: Diagrama de um *perceptron*

Neste caso, é possível intuir que redes mais complexas de *perceptrons* podem tomar decisões mais complicadas. A Figura 3.2 mostra uma rede desse tipo. Nessa, a primeira camada de *perceptrons* é responsável por ponderar, de maneiras diferentes, os valores da entrada. Na segunda camada, os *perceptrons* ponderam sobre os resultados da primeira camada e, com isso, são capazes de tomar decisões em um nível mais abstrato. Por fim, a última camada recebe essas decisões mais complexas da segunda camada e produz uma única saída, tornando o processo de tomada de decisão dessa rede, em relação à um único *perceptron*, mais sofisticada.

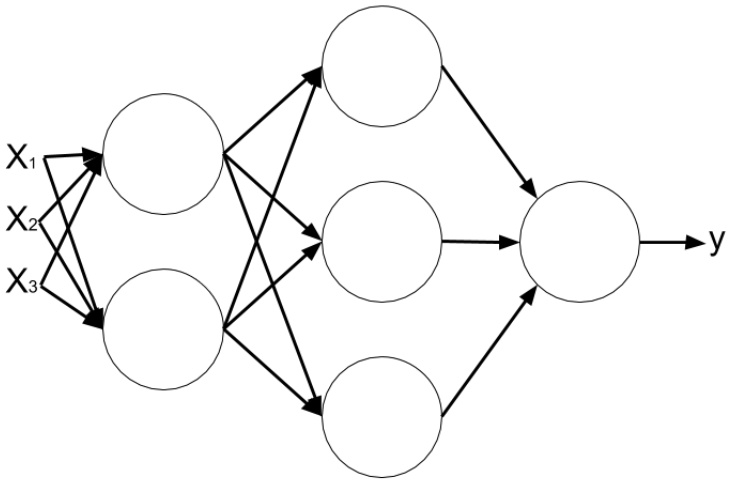

Figura 3.2: Um exemplo de rede complexa de *perceptrons*

3.2 Neurônio sigmóide e *Multi Layer Perceptron*

Quando se realiza o treinamento usando o Método de Descida Gradiente, é necessário que a função de custo J seja diferenciável, uma vez que o método depende das derivadas parciais dessa em relação aos parâmetros (no caso de RNA, os pesos e o corte). Caso J seja diferenciável, é possível calcular o gradiente e, com isso, ajustar gradativamente os parâmetros. O problema de RNA's com *perceptrons* é que a função que aplica o corte, conhecida como função degrau, não é diferenciável, tornando J não diferenciável. Quando isso ocorre, pequenos ajustes nos parâmetros podem: (i) não modificar os valores gerados na saída, ou (ii) modificar a saída abruptamente, não permitindo o ajuste gradativo dos parâmetros. Isso dificulta o treinamento, uma vez que se torna difícil controlar o comportamento dos parâmetros e consequentemente, das saídas produzidas pelo modelo. Para contornar esse problema, foi criado um novo tipo de neurônio, chamado de neurônio sigmóide.

Neurônios sigmóides são *perceptrons* modificados de maneira a permitir a aprendizagem gradativa. A diferença entre eles está no cálculo de saída. Enquanto que os *perceptrons*, a partir do valor de z, definem uma saída binária aplicando a função degrau, o neurônio sigmóide produz, em sua saída, um valor real, o qual é gerado por uma função sigmóide. A forma algébrica da função sigmóide é apresentada na Equação 3.2.

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}}. \tag{3.2}$$

A Figura 3.3 apresenta a forma gráfica das funções sigmóide e degrau respectivamente. Pode-se verificar que a função sigmóide é apenas uma versão suavizada da função degrau. Porém, usando a função sigmóide, a qual é diferenciável, é possível utilizar-se do Método da Descida de Gradiente em J e, com isso, realizar um aprendizado gradativo. A função sigmóide não é a única que permite a aplicação do Método de Descida de Gradiente em J. Funções não lineares diferenciáveis, aplicadas ao valor de z, são chamadas *funções de ativação*.

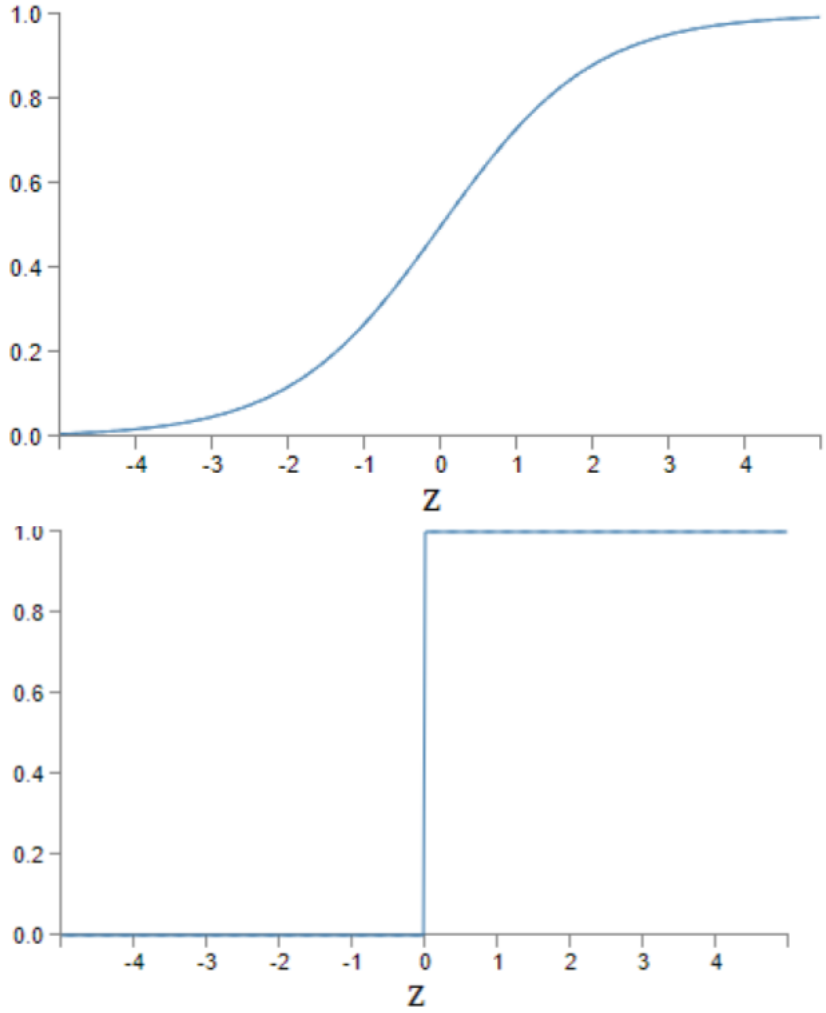

Figura 3.3: Respectivamente, a função sigmóide e a função degrau, extraído de [2]

Uma RNA composta por três ou mais camadas de neurônios é denominada *Multi Layer Perceptron* (MLP). Nesta arquitetura, a camada que recebe as entradas é chamada de *camada de entrada*, enquanto que os neurônios componentes da camada são chamados de *neurônios de entrada*. A mesma regra vale para a camada intermediária, chamada de *camada escondida*, sendo que um MLP pode ter mais de uma camada desse tipo, e para a camada que produz a saída, chamada de *camada de saída*. No caso, os neurônios de

entrada são diferentes dos neurônios escondidos e de saída. Na camada de entrada, há um neurônio para cada atributo de entrada, sendo que o valor de saída do neurônio n_i é igual ao atributo de entrada x_i. Quanto às outras camadas, os neurônios são neurônios sigmóides.

Algoritmo 1: Aprendizagem (RNA, Θ, X, Y)

Entrada: Uma RNA com N camadas
Dados: O conjunto de parâmetros Θ da RNA
Dados: O conjunto de observações X e suas variáveis-alvo Y
Resultado: Parâmetros Θ otimizados
Inicializa-se os parâmetros Θ de acordo com um método de inicialização;
para *(contagem ← 1 até 100)* **faça**
`/* Propagação:` `*/`
para *cada* x_i, y_i *em* (X, Y) **faça**
para *neurônio i na camada 1* **faça**
$saída_i^{(1)} \leftarrow x_i$;
fim
para $n \leftarrow$ *2 à N-1* **faça**
para *neurônio j na camada n* **faça**
$\text{somatório}_j^{(n)} \leftarrow \sum_i \theta_{i,j}^{(n)}.saída_i^{n-1}$;
$saída_j^{(n)} \leftarrow \sigma(somatório_j^{(n)})$;
fim
fim
fim
`/* Retropropagação:` `*/`
para *neurônio j na camada N* **faça**
$\gamma_j^{(N)} \leftarrow \sigma'(somatório_j^{(N)}).(y_j - saída_j^{(N)})$;
fim
para $n \leftarrow$ *N-1 até 2* **faça**
para *neurônio j na camada n* **faça**
$\gamma_j^{(n)} \leftarrow \sigma'(somatório_j^{(n)}) \sum_i \theta_{i,j}^{(n+1)}.\gamma_i^{(n+1)}$;
fim
fim
para $\theta_{i,j}^{(n)}$ *em* Θ, *onde n = 2, 3, ..., N* **faça**
$\theta_{i,j}^{(n)} \leftarrow \theta_{i,j}^{(n)} + \alpha.\gamma_j^{(n)}$;
fim
fim
Calcula-se J(Θ);

3.2.1 Aprendizagem em MLP's

Suponha um MLP com N camadas, um conjunto de parâmetros Θ, onde $\theta_{i,j}^{(n)}$ é o parâmetro aplicado à saída do neurônio i da camada $n-1$, a qual é usada como entrada no neurônio j da camada n, a função sigmóide σ, apresentada na Equação 3.2, e uma taxa de aprendizagem α fixa. O Algoritmo 1 apresenta o pseudocódigo do algoritmo de aprendizagem, o qual realiza 100 iterações de modificação dos parâmetros usando o Método da Descida de Gradiente. Durante o treinamento, duas etapas ocorrem: A propagação e a retropropagação. A propagação consiste no cálculo das saídas, a partir das observações da base de treino (i.e. X e Y). Durante esse cálculo os valores partem da entrada, passam pelas camadas escondidas e, por fim, calcula-se os valores finais na camada de saída, $saída_i^{(N)}$.

Calculado a derivada parcial de J, em relação à cada parâmetro em Θ, na etapa de retropropagação, é possível ajustá-los, a fim de minimizar J (i.e., aplicar o Método da Descida de Gradiente). No caso, os ajustes dos parâmetros, γ_i^n, são calculados no sentido inverso à propagação, começando com os neurônios de saída e seguindo para os neurônios das camadas escondidas.

3.3 Aprendizagem profunda

Como discutido na Seção 3.1, redes neuronais mais complexas são capazes de gerar melhores performances em problemas mais sofisticados. Porém, uma das dificuldades de se treinar redes neuronais com grandes quantidades de camadas e neurônios é a demanda computacional que essas exigem. Recentemente, o uso de unidades de processamento gráfico (GPU) possibilitaram um ganho de performance, em termos de tempo, no treinamento de tais redes e permite que esse seja realizado em tempo hábil. Com isso, tornou-se possível treinar redes neuronais com múltiplas camadas escondidas, denominadas redes neuronais profundas.

As seções a seguir discutem alguns tipos de redes neuronais profundas, as quais são relevantes para este trabalho.

3.3.1 Redes neuronais convolucionais

Rede Neuronal Convolucional (CNN) são RNA's especializadas em processamento de dados que possuam uma topologia de grade (i.e., séries temporais, imagens, áudio, vídeo) [2]. O nome dado a esse tipo de RNA decorre do fato dessas aplicarem operações de convolução em, pelo menos, uma de suas camadas.

Convolução discreta

Em se tratando de dados de imagem, em redes neuronais convolutivas, aplica-se um tipo especial de convolução: A convolução discreta em duas dimensões.

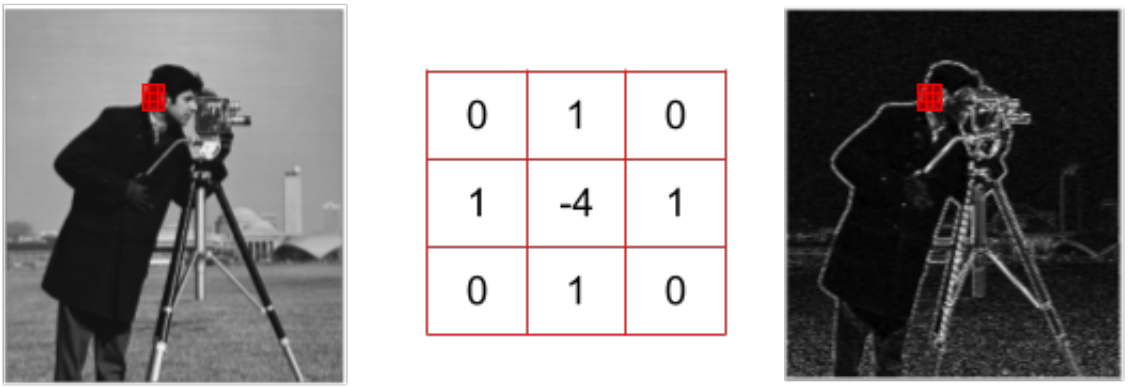

Figura 3.4: Demonstração de aplicação de convolução

A Figura 3.4 mostra a aplicação de convolução em uma imagem. Nessa, a partir de uma imagem original NxM à esquerda, aplica-se um filtro 3x3, o qual produz uma versão diferente da imagem original na saída. O filtro é representado pelos quadrados vermelhos. Os parâmetros do filtro, apresentados ao centro da Figura 3.4, são os fatores multiplicativos aplicados aos *pixels* da imagem de maneira *element-wise*. Após aplicar as multiplicações, soma-se esses valores e substitui-se o *pixel* central pelo valor calculado. Realiza-se essa operação em todos os pixels da imagem. A imagem à direita apresenta o resultado normalizado da convolução. No caso, os parâmetros associados à esse filtro permitem que esse detecte bordas na imagem. Os valores componentes dos filtros, em uma CNN, são parâmetros da rede, os quais são aprendidos durante o treinamento usando o Método da Descida de Gradiente. Dessa maneira, a CNN é capaz de realizar transformações nos dados, a medida que esses são processados.

Convoluções introduzem três propriedades que podem melhorar um sistema de aprendizagem de máquina: **interações esparsas**, **compartilhamento de parâmetros** e **representações equivariantes** [2].

Em redes neuronais que não usam convolução, é necessário **um parâmetro para cada entrada**. No contexto de processamento de imagens, seria necessário **um parâmetro para cada pixel**. Devido às interações esparsas, redes neuronais convolucionais apresentam filtros, em suas camadas convolutivas, de dimensões menores que as imagens de entrada. Durante o processamento, é possível detectar pequenas características das imagens por meio desses filtros (i.e. bordas, gradientes, linhas). Isso demanda uma

quantidade de parâmetros menor, diminuindo os requisitos de memória e o número de operações para calcular as saídas. Compartilhamento de parâmetros se refere ao uso de um mesmo parâmetro no cálculo de várias funções. Em outras redes neuronais, cada peso é usado apenas uma vez, multiplicando um elemento da entrada, no cálculo da saída de uma camada. Em uma Rede Neuronal Convolucional, um parâmetro do filtro é usado em todos os valores da entrada. Neste sentido, ao invés de se aprender um conjunto separado de parâmetros para cada valor de entrada, aprende-se apenas um conjunto de parâmetros para todas as entradas. No caso das CNN's, essa forma de compartilhamento de parâmetros leva à RNA ter a propriedade de representações equivariantes. Isso significa que as representações internas ao modelo capturam as propriedades da imagem e são robustas. Por isso, as representações internas são invariantes às pequenas modificações da entrada (e.g., diferenças de iluminação).

Tipos de camada em uma CNN

As camadas mais comuns em uma rede neuronal convolutiva são a camada de convolução, a camada de *pooling* e a camada totalmente conectada, também conhecida como camada densa. A camada de convolução é responsável por aplicar as operações de convolução em uma CNN. Nessa, aplica-se N filtros na entrada, gerando N valores na saída, chamados *mapas de ativação*. Com isso, camadas convolutivas são capazes de identificar características que compõem a imagem (i.e. presença de borda, linhas, gradientes, ângulos). A camada de *pooling*, normalmente, segue uma camada convolutiva e realiza uma subamostragem dos valores nos mapas de ativação. Por fim, a camada totalmente conectada, também conhecida como camada densa, é equivalente a um MLP. Essa é usada para combinar os atributos gerados pelas outras camadas e produzir a saída da CNN. A Figura 3.5 apresenta a arquitetura de uma CNN contendo 2 camadas convolutivas e 1 camada densa.

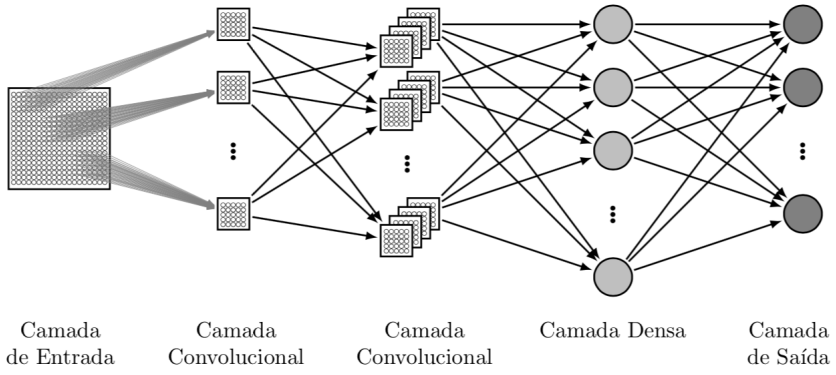

Figura 3.5: Um exemplo de CNN, retirado de [3]

Uma arquitetura de Rede Neuronal Convolutiva mais complexa é a *GoogleNet* [32]. Essa possui 27 camadas, dentre as quais: 11 apresentam convoluções, 5 são camadas de *pooling* e 1 camada é totalmente conectada.

3.3.2 Rede Neuronal Recorrente

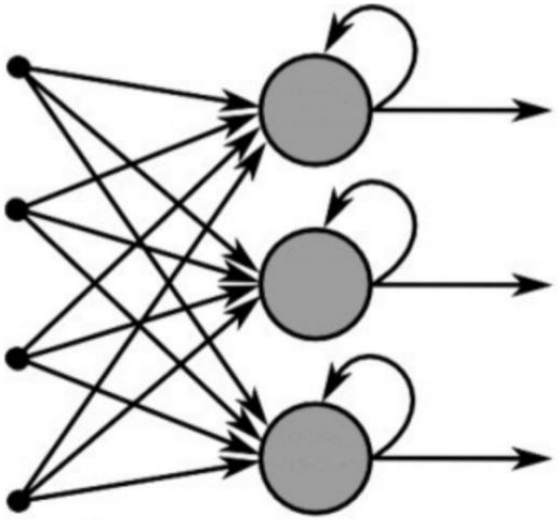

Figura 3.6: Diagrama de fluxo de informação em uma RNN. Extraído de [4]

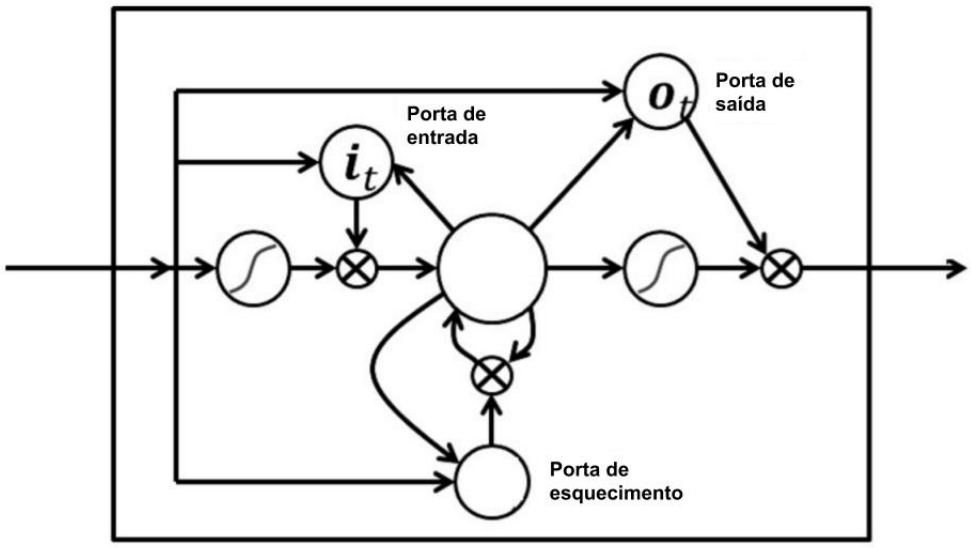

Figura 3.7: Uma célula LSTM, modificada de [4]

Rede Neuronal Recorrente (RNN) é uma família de RNA's destinadas a processamento de sequências. A RNN foi criada nos anos 80, mas exigia um poder computacional muito

alto. Seu potencial só começou a ser explorado recentemente, devido aos avanços em performance dos computadores, a quantidade massiva de dados que se tem acesso atualmente e o uso de Long Short-Term Memory [33] a partir dos anos 90. Uma das características mais importantes das RNN's é sua **memória interna**, o que a torna eficiente em tarefas envolvendo dados sequenciais e dependências entre as entradas (e.g. dados de finanças, séries temporais, vídeo). Diferente das redes neuronais apresentadas nas seções anteriores, as quais fazem parte de um grupo de redes neuronais chamados de *feedforward networks*, as informações dentro de uma RNN passam por ciclos. Com isso, quando realizado um cálculo de saída, a RNN leva em consideração a entrada atual e o que foi aprendido com as entradas passadas. Neste contexto, é dito que uma RNN possui uma **memória de curto prazo**. Essa propriedade é importante, pois, em dados sequenciais, entradas passadas possuem informação crucial para as próximas entradas. O diagrama da Figura 3.6 demonstra o fluxo de informação em uma camada da RNN, sendo que a retropropagação representa que as entradas anteriores têm efeito no cálculo da nova saída.

Uma das dificuldades, verificada empiricamente [34], encontrada por RNN's é: Alguns problemas sequenciais apresentam **dependências de longo prazo**, ou seja, entradas distantes na sequência possuem informação relevante para o cálculo da saída corrente. No caso, a memória das RNN's não permite a modelagem de tais dependências. Com isso, foi desenvolvida um novo tipo de rede neuronal, a qual é uma extensão das RNN's: As redes *Long Short-Term Memory* (LSTM). Para permitir que RNN's modelem dependências de longo prazo, LSTM's adicionaram **unidades de memória explícitas** às RNN's. Essas unidades de memória podem ser entendidas como células com portas, as quais podem armazenar ou deletar informação (abrir ou não a porta), de acordo com a importância dessa. Com a adição dessa célula, RNN's aprendem a identificar quais informações na sequência são importantes para a entrada corrente e quais não.

A Figura 3.7 apresenta uma célula LSTM, a qual apresenta 3 portas: Entrada, saída e esquecimento. A porta de entrada define se uma entrada deve, ou não, ser armazenada, a porta de esquecimento decide se um valor armazenado deve ser removido e a porta de saída decide se o valor armazenado deve impactar a saída atual [33].

Atualmente, diversas pesquisas em aprendizagem profunda vêm sendo realizadas, a fim de descobrir novas aplicações e arquiteturas. Tome como exemplo um dos trabalhos mais recentes, em processamento de linguagem natural, realizado por Jeremy Howard et al. [35], o qual criou um modelo de linguagem universal (ULMFiT) pré-treinado capaz de realizar transferência de aprendizagem em qualquer tarefa em processamento de linguagem natural, obtendo, no momento de sua publicação, os melhores resultados do estado da arte em seis tarefas de classificação de texto. Além disso, por se tratar de um modelo pré-treinado, é capaz de obter resultados significativos em diversas tarefas com apenas 100

exemplos de treinamento.

Nos Capítulos 2 e 3, conceitos fundamentais sobre aprendizagem de máquina e redes neuronais foram introduzidos, uma vez que esses apresentam relevância ao entendimento da metodologia proposta no Capítulo 5. O capítulo seguinte realiza uma revisão dos trabalhos relacionados a sumarização de vídeo, destacando abordagens relevantes para este trabalho.

Capítulo 4

Trabalhos Relacionados

Técnicas em sumarização automática de vídeos apresentam duas abordagens principais: Não-supervisionadas e supervisionadas, as quais serão revisadas nas seções deste capítulo. Em abordagens supervisionadas, aprofunda-se a revisão no trabalho de Zhang et al. [5], uma vez que esse apresenta grande relevância para o entendimento deste trabalho.

4.1 Abordagens não supervisionadas

As abordagens não-supervisionadas utilizam um critério manual para realizar a definição de escores de importância e selecionar subconjuntos de quadros a partir dos vídeos. Dentre os critérios informativos, inclui-se relevância [16], importância [12] e diversidade [8].

Jiang Zhang et al. [8] apresentou, em 1997, uma solução para parseamento, recuperação e navegação baseada em conteúdo de vídeo. O parseamento segmenta e abstrai o vídeo original e a recuperação e navegação do vídeo é baseada em quadros-chave, atributos temporais e atributos de movimento dos segmentos. Por fim, Jiang Zhang et al. apresenta em detalhes as funções do sistema integrado produzido durante o trabalho, o qual realiza a segmentação e facilita a recuperação e navegação em vídeos.

Em 2002, Ma et al. [20] apresentou uma estrutura genérica de sumarização de vídeos baseada na modelagem da atenção dos espectadores. Usando essa estrutura, a qual não leva em consideração o entendimento semântico dos conteúdos dos vídeos, aproveitou-se de modelos computacionais de atenção e eliminou-se a necessidade de heurísticas complexas na sumarização automática.

Usando modelagem de grafos, Chong-Wah Ngo et al. [13], em 2003, propôs uma abordagem para sumarizar automaticamente baseada na análise das estruturas e destaques dos vídeos. Por meio de um algoritmo de corte normalizado, otimizou-se a partição global dos quadros em agrupamentos. Com isso, usou-se um modelo de atenção de movimento baseado em percepção humana para calcular a qualidade de percepção desses agrupa-

mentos. Em seguida, com os agrupamentos e valores de atenção, formou-se um grafo temporal que descreve a evolução e a importância perceptiva dos agrupamentos do vídeo. Por fim, o grafo é utilizado para agrupar agrupamentos similares em cenas, enquanto que as importâncias perceptuais são utilizadas para selecionar as melhores cenas para o sumário.

Em 2006, Mundur et al. [10] propôs uma técnica de sumarização automática de vídeos baseada em quadros-chave [36]. Para isso, representou-se os quadros como vetores multidimensionais e utilizou-se a Triangulação de Delaunay para agrupá-los de acordo com suas características visuais.

Richang Hong et al. [16], em 2009, apresentou uma solução baseada em minerar e organizar cenas chave em vídeos acompanhados de marcadores (*tags*) presentes na *internet*. No processo de sumarização proposto, primeiramente, realizou-se uma busca de vídeos associados com seus marcadores. Em seguida, cenas-chave foram estabelecidas usando detecção de quadros quase duplicados, as quais são classificadas de acordo com sua informatividade e organizadas em ordem cronológica. Por fim, sumários foram formulados baseados nas relevâncias das cenas-chave e no tamanho desejado para o sumário.

4.2 Abordagens supervisionadas

Como abordagens supervisionadas são capazes de usar dados anotados, essas são mais adequadas para sumarizar vídeos de maneira alinhada à maneira com que humanos fariam.

Gygli et al. [23] propôs, em 2014, uma abordagem e uma base de referência para sumarização de vídeos. Em sua abordagem, inicialmente, segmentou-se o vídeo usando o método de segmentação de *superquadros*. Em seguida, estimou-se o grau de interesse do *superquadro* e selecionou-se um subconjunto desses, a fim de criar um sumário informativo e interessante. Por, fim, Gygli et al. introduziu uma base referência para trabalhos em sumarização automática de vídeos: Um conjunto de dados anotados, adquiridos em experimentos controlados, permitindo, assim, que trabalhos futuros possam realizar comparações de resultados entre si.

Em 2015, Gygli et al. [21] apresentou um método de sumarizar vídeos capturados casualmente, o qual cria sumários curtos, otimizando o grau de interesse e o grau de representação desses. Por meio de uma abordagem supervisionada, aprendeu-se as características globais de um bom sumário. Por fim, otimizou-se a criação do sumário a fim de comparar os sumários gerados com a referência proposta em [23].

Além dos trabalhos discutidos, outros usaram Ponto de Processo Determinante (DPP), um modelo probabilístico que caracteriza diversidade em subconjuntos de quadros dos vídeos, e obtiveram resultados expressivos.

Em 2014, Gong et al. [9] tratou o problema da sumarização de vídeos automática como um problema de seleção de subconjunto supervisionada. Na abordagem, o sistema proposto aprendeu, a partir de sumários criados por humano, a selecionar subconjuntos informativos e diversos, de maneira a obter melhores resultados utilizando as métricas de avaliação da referência. Para isso, propôs-se o modelo *sequential determinantal point process* (seqDPP), um modelo baseado em DPP.

Além do trabalho de Gong et al., Chao et al. propôs, em 2015, uma técnica de estimação de parâmetro baseada no princípio de separação de margens grandes para DPP, melhorando a capacidade desse modelo de selecionar subconjuntos diversos de quadros. Os resultados obtidos por Chao et al. podem ser aplicados em sumarização de vídeos e de documentos.

Por fim, em 2016, Zhang et al. [22] propôs um método de seleção diversificada de quadros, de maneira a empregar supervisão, utilizando anotações humanas, e criar sumários de vídeos baseados em quadros. Em seguida, Zhang et al. demonstrou como transformar as sumarizações baseadas em quadros em sumarizações baseadas em cenas, tornando-as assistíveis.

No contexto supervisionado, notou-se que a sumarização de vídeos é um problema de dados sequenciais, os quais apresentam **dependências de longo alcance**. Com isso, o trabalho de Zhang et al. [5], em 2016, foi o primeiro a utilizar LSTM's, a fim de modelar as dependências temporais dos vídeos. A seção a seguir é dedicada a explicar o trabalho realizado por Zhang et al.

4.3 Sumarização de vídeo usando *Long Short-Term Memory*

Em seu trabalho, Zhang et al. propôs uma nova técnica para sumarizar vídeos automaticamente, por meio da seleção de quadros-chave, ou cenas-chave. Tratando o problema de sumarização como uma tarefa de predição estruturada, utilizou-se de LSTM's para modelar as dependências temporais de extensão variada entre os quadros, de maneira a gerar um sumário representativo e compacto. Neste contexto, desenvolveu-se dois modelos que levam em consideração a estrutura sequencial dos vídeos: vsLSTM, o qual utiliza duas camadas LSTM e um MLP para gerar importâncias dos quadros, e dppLSTM, o qual melhora o modelo vsLSTM, adicionando um MLP e uma camada DPP, produzindo uma seleção representativa e diversa de quadros em sua saída.

A seção a seguir explica o modelo DPP, a fim de demonstrar como esse obtém diversidade na seleção de quadros.

4.3.1 Ponto de Processo Determinante (DPP)

Dado um conjunto Z com N quadros, os quais compõem um vídeo, e uma matriz *kernel* **L**, NxN, que armazena as relações de similaridade pareada entre os quadros, o modelo DPP codifica a probabilidade de amostrar qualquer subconjunto de Z [9].

A probabilidade de um subconjunto z é proporcional ao determinante da menor principal L_z, ou seja, a matriz original L contendo apenas as linhas e colunas dos quadros presentes no subconjunto z, como demonstrado na Equação (4.1).

$$\mathbf{P}(\mathbf{z} \subset \mathbf{Z}; \mathbf{L}) = \frac{det(\mathbf{L_z})}{det(\mathbf{L} + \mathbf{I})}, \tag{4.1}$$

em que **I** é a matriz identidade de tamanho N. Com esse cálculo de probabilidade, se quadros idênticos aparecerem no mesmo subconjunto, L_z terá linhas e colunas iguais, levando o determinante a 0, ou seja a probabilidade desse subconjunto seria 0. Uma probabilidade alta seria capturada em subconjuntos com alta dissimilaridade entre quadros. Com isso, usar o modelo DPP, no contexto de sumarização de vídeos, permite modelar as similaridades entre quadros e, assim, possibilita gerar sumários com maior diversidade.

4.3.2 Modelos vsLSTM e dppLSTM

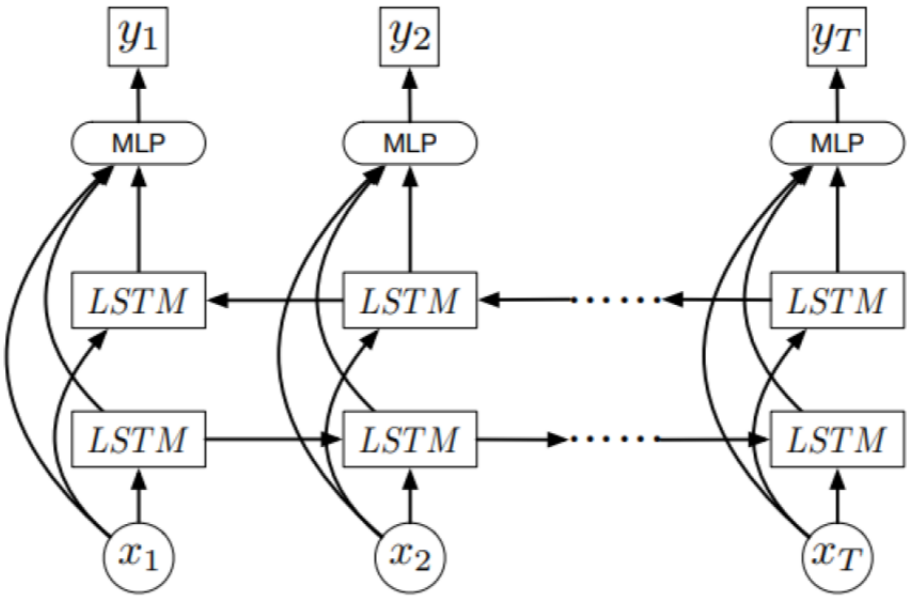

Figura 4.1: O modelo vsLSTM, extraído de [5]

O modelo vsLSTM, apresentado na Figura 4.1, é composto por duas camadas LSTM, também chamadas conjuntamente de LSTM bidirecional [37], de maneira que a primeira camada é responsável por modelar as dependências sequenciais do vídeo na direção para a frente, enquanto que a segunda camada modela as dependências sequenciais no sentido contrário. As entradas do modelo são atributos visuais profundos extraídos do quadro x_i. Para produzir os escore de importância de um quadro, a saída combina os resultados obtidos na LSTM bidirecional e os atributos visuais profundos do quadros usando um *Multi Layer Perceptron*. Esse calcula o escore de importância do quadro y_i. No treinamento deste modelo, utiliza-se bases de treinamento contendo os atributos visuais profundos de quadros, como entradas, e escores de importância dos quadros, como valores alvo. Para otimizar os parâmetros do modelo é empregado o Método da Descida de Gradiente Estocástico [38], uma variação do Método de Descida de Gradiente.

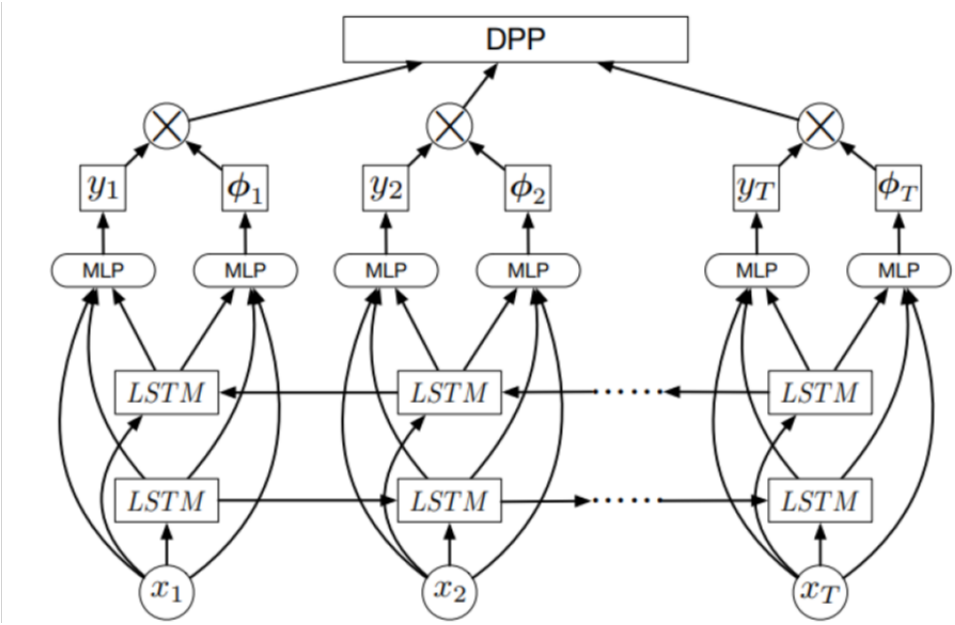

Figura 4.2: O modelo dppLSTM, extraído de [5]

No modelo dppLSTM, apresentado na Figura 4.2, melhora-se o modelo vsLSTM, por meio da modelagem de similaridade pareada dos quadros usando DPP. Com isso, o modelo melhora a diversidade na seleção de quadros, eliminando quadros redundantes. Esse modelo usa dois MLP's: O primeiro, semelhante ao MLP do modelo vsLSTM, é responsável por gerar a importância dos quadros y_i, enquanto que o segundo gera ϕ_i. A Equação (4.2) demonstra como, a partir dos valores de saída dos MLP's, calcula-se as entradas da matriz *kernel* L, usada no modelo DPP:

$$\mathbf{L_{tt'}} = y_t y_{t'} \phi_t^T \phi_{t'} \tag{4.2}$$

Ou seja, a entrada $L_{tt'}$ da matriz *kernel* para a similaridade entre os quadros x_t e $x_{t'}$ é igual ao produto interno entre os vetores ϕ_t e $\phi_{t'}$, vetores de saída do segundo MLP, ponderado pelos escores de importância dos quadros.

Treinamento do modelo dppLSTM

Na aprendizagem dos parâmetros do modelo dppLSTM, utilizou-se uma rotina em estágios: No primeiro estágio, treina-se a primeira MLP, responsável por gerar os escores de importância, otimizando o erro de predição desses. O segundo estágio otimiza todos os parâmetros do modelo (MLP's e LSTM's). A otimização é baseada na máxima verossimilhança (MLE) especificada pela DPP. No caso, seja $z* \in Z$ o subconjunto de quadros-chave alvo para um vídeo, os parâmetros θ no modelo são otimizados segundo a Equação 4.3:

$$\theta^* = argmax_\theta \sum_i logP(z^{(i)*} \in Z^{(i)}; L^{(i)}(\theta)) \tag{4.3}$$

Onde i indexa o subconjunto alvo $z*$, o conjunto de quadros Z e a matriz *kernel* L do i-ésimo vídeo. Otimiza-se θ usando o Método da Descida de Gradiente Estocástico.

O subconjunto de quadros-chave, produzidos na saída do modelo dppLSTM, é calculado, a partir da matriz *kernel* e dos escores de importância, usando inferência aproximada MAP [39]. Esse algoritmo é utilizado em problemas de maximização submodular irrestrita, no qual, dada uma função $f : 2^Z \rightarrow \mathbb{R}^+$, deseja-se encontrar um subconjunto $z \subseteq Z$ que maximize $f(z)$. No caso do modelo dppLSTM, f é a função de probabilidade calculada por DPP, o conjunto Z é o conjunto de quadros e o subconjunto z é o subconjunto de quadros-chave.

Nas bases de teste, os alvos estão codificados como sumários baseados em cenas-chave, ou seja, esses são compostos por intervalos de tempo contínuos extraídos do vídeo original. Como o modelo dppLSTM gera, em sua saída, subconjuntos de quadros-chave, os quais são discretos no tempo, é necessário, para avaliar o modelo, realizar uma conversão da sua saída. Para converter quadros-chave em cenas-chave, primeiramente, segmenta-se o vídeo em intervalos disjuntos usando Segmentação Temporal de *Kernel* (KTS), uma ferramenta estatística capaz de detectar mudanças entre quadros, possibilitando descobrir onde uma cena começa e termina [40]. Em seguida, caso uma cena tenha, pelo menos, um quadro-chave, marca-a como cena-chave. Em seguida, classificam-se as cenas-chave de acordo com o número de quadros-chave presentes e, em ordem decrescente, elas são adicionadas ao sumário até um limite de tempo pré-estabelecido seja alcançado (e.g., usando o algoritmo da mochila [41]). Caso o limite de tempo não seja alcançado, seleciona-se outras cenas

usando, como critério de seleção, a média dos escores de importância dos segmentos até alcançá-lo.

A partir da contextualização do problema, apresentada no Capítulo 1, dos conceitos teóricos fundamentais sobre aprendizagem de máquina e redes neuronais, apresentados nos Capítulos 2 e 3, e da revisão dos trabalhos relacionados, com foco no trabalho sobre sumarização usando LSTM, discutidos neste capítulo, explica-se a proposta deste trabalho no Capítulo 5, a partir da qual busca-se atingir os objetivos apresentados na seção 1.3.

Capítulo 5

Metodologia Proposta

A partir disso, este trabalho introduz **o conceito de estilo em sumarização de vídeo baseado em critérios de seleção de cenas e propõe novos critérios de seleção de cena (i.e., Maior ou Igual a Média, Menor que a Média, Maior ou igual a Média + 2.Desvio Padrão e Randômico Ponderado)**. O diagrama da Figura 5.1 apresenta uma visão geral da metodologia proposta.

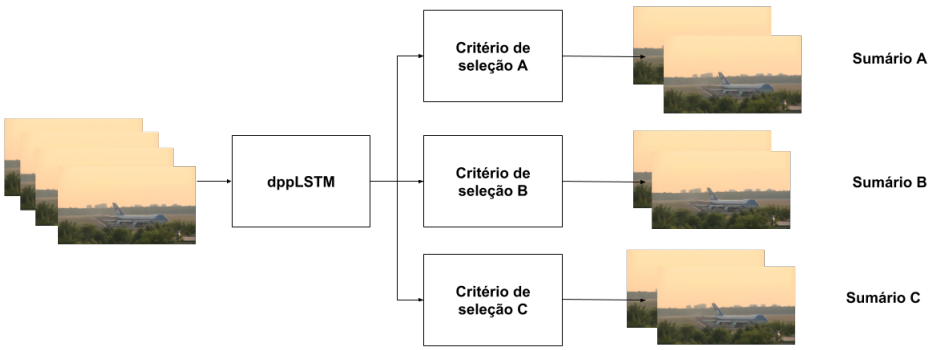

Figura 5.1: Diagrama da visão geral da metodologia proposta

Primeiramente, essa seção apresenta o conceito de estilo de sumarização. Em seguida, expõe as bases de dados utilizadas, a configuração de dados empregada no treinamento do modelo dppLSTM e a forma com que foram gerados os atributos dos quadros dos vídeos. Com isso, apresenta-se como o modelo dppLSTM foi utilizado para gerar os escores de importância das cenas. Em seguida, discutem-se as atividades de manipulação de quadros realizadas, a fim de possibilitar, de fato, a geração de sumários assistíveis. Por fim, definem-se os critérios de seleção, baseados no conceito de estilo, propostos.

5.1 Estilo de sumarização

Segundo o dicionário Dicio [42], dentre outros significados, estilo é **um conjunto de qualidades características de um objeto**. Aproveitando deste significado, definimos estilo de sumarização como: Dado um video e seus sumários, gerados usando técnicas de sumarização, o estilo de um sumário A são as características **visuais** e **sequenciais** que são próprias dele, diferindo-o dos outros possíveis sumários.

Usando este novo conceito, pode-se definir o trabalho realizado como: A criação de critérios de seleção de cenas, baseados nos escores de importância delas, gerados pelo modelo dppLSTM, os quais produzem sumários com **estilos de sumarização diferentes**.

5.2 Bases de dados

No treinamento do modelo dppLSTM, realizado por Zhang et al. [5], foram utilizadas quatro bases de dados no total: SumMe [23], TVSum [41], OVP [43, 44] e Youtube [43], sendo que as últimas duas foram usadas apenas para aumentar a quantidade de dados de treinamento e validação, não sendo usadas para teste. As seguintes configurações foram experimentadas no aprendizado do modelo (a Tabela 5.1 apresenta de forma clara as diferentes configurações):

- **Canônica:** Configuração padrão em treinamento supervisionado, em que a base de treino, validação e teste são da mesma base de dados, mas disjuntas.
- **Aumentada:** Para uma base de dados (SumMe ou TVSUMM), seleciona-se randomicamente 20% dos vídeos para teste. Os outros 80% são agrupados com as outras três bases de dados para formar as bases de treinamento e validação.
- **Transferência:** Considera uma base de dados de referência (SumMe ou TVSUMM), usa-se ela inteiramente para teste. As outras três bases são usadas para treino e validação.

Tabela 5.1: Configurações de treinamento e teste

Bases de dados	Configurações	Treino e Validação	Teste
SumMe	Canônica	80% SumMe	20% SumMe
	Aumentada	OVP + Youtube + TVSum + 80% SumMe	20% SumMe
	Transferência	OVP + Youtube + TVSum	SumMe
TVSum	Canônica	80% TVSum	20% TVSum
	Aumentada	OVP + Youtube + SumMe + 80% TVSum	20% TVSum
	Transferência	OVP + Youtube + SumMe	TVSum

Tabela 5.2: F-Scores obtidos na base SumMe usando dppLSTM

Configurações	F-Score
Canônica	38.6 ± 0.8
Aumentada	42.9 ± 0.5
Transferência	41.8 ± 0.5

A Tabela 5.2 apresenta os resultados obtidos usando a base SumMe como teste. No caso, Zhang et al. obteve os melhores resultados do estado da arte usando as configurações *Aumentada* e *Transferência*.

Neste trabalho, decidiu-se utilizar o modelo treinado empregando transferência (SumMe como base de teste). Essa escolha foi feita devido a disponibilidade dos parâmetros pré-treinados, em [26], usando essa configuração, o que facilitou a aplicação do modelo e removeu a necessidade de treiná-lo. Mantendo a lógica da configuração, a base SumMe foi a escolhida para se aplicar os critérios de seleção.

Figura 5.2: Quadro-exemplo do vídeo *Air Force One* da base SumMe

5.2.1 Atributos dos quadros

No trabalho de Zhang et al., foram comparados métodos profundos e superficiais de geração de atributos para os quadros. Os métodos superficiais são aqueles que não usam Redes Neuronais Artificiais para sua geração (i.e., histogramas de cores, GIST, HOG, dense SIFT) enquanto que métodos profundos fazem uso de redes neuronais artificiais da seguinte maneira: (i) Define-se um modelo (arquitetura) de Rede Neuronal Artificial profunda, (ii) Treina-se o modelo para executar uma tarefa (i.e., classificar imagens), (iii) Com o modelo treinado, aplica-o em uma imagem, porém, ao invés de pegar a saída final do modelo, pega-se uma das representações abstratas intermediárias (geralmente, contendo muitas dimensões) e a usa como vetor de atributos da imagem. A Figura 5.3

apresenta, de maneira clara, um exemplo de local para extração de atributos profundos em uma CNN genérica treinada para classificação binária.

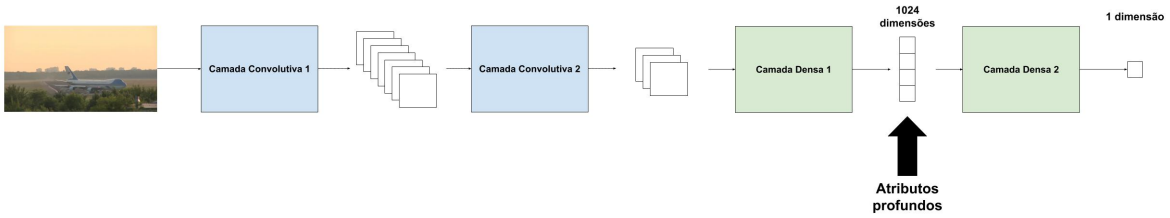

Figura 5.3: Exemplo de extração de atributos profundos de uma imagem em uma CNN

No caso, para a geração de atributos profundos de cada quadro, extraiu-se a saída da penúltima camada (*pool* 5) do modelo GoogleNet [32], apresentando 1024 dimensões. Nas análises realizadas no trabalho de Zhang et al., concluiu-se que usar atributos profundos resultam em uma maior acurácia do modelo [5], logo, neste trabalho, decidiu-se por manter o uso deles. Os dados de atributos profundos dos vídeos foram disponibilizados em [26] no formato de banco de dados HDF5 (Hierarchical Data Format version 5), um padrão otimizado para armazenamento de grandes volumes de dados em um único arquivo, logo, não foi necessário gerar os atributos profundos novamente.

5.3 Gerando os escores de importância das cenas

Como visto no Capítulo 4, o modelo dppLSTM gera três saídas: O subconjunto de quadros-chave (a saída principal do modelo), os escores de importância dos quadros e a matriz *kernel* de diversidade. A partir destas saídas, Zhang et al. aplica seu critério de seleção e identifica quais cenas devem entrar no sumário. Uma das etapas do critério de seleção é gerar o escore de importância das cenas dos vídeos. Isso é realizado fazendo, **para cada cena, a média dos escores de importância dos quadros que compõem uma cena**. A Figura 5.4 apresenta esse processo de geração dos escores das cenas.

Neste sentido, aplicou-se o modelo dppLSTM e gerou-se os escores de importância das cenas de **todos os vídeos da base SumMe**, armazenando os resultados em formato *pickle* [45].

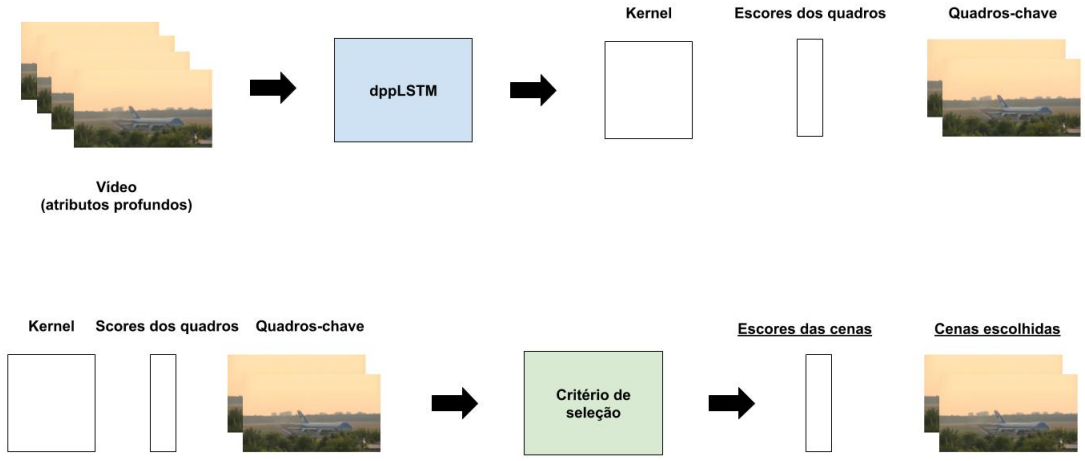

Figura 5.4: Diagrama de extração das importâncias das cenas e da seleção de cenas de Zhang et al.

5.4 Geração de sumários assistíveis

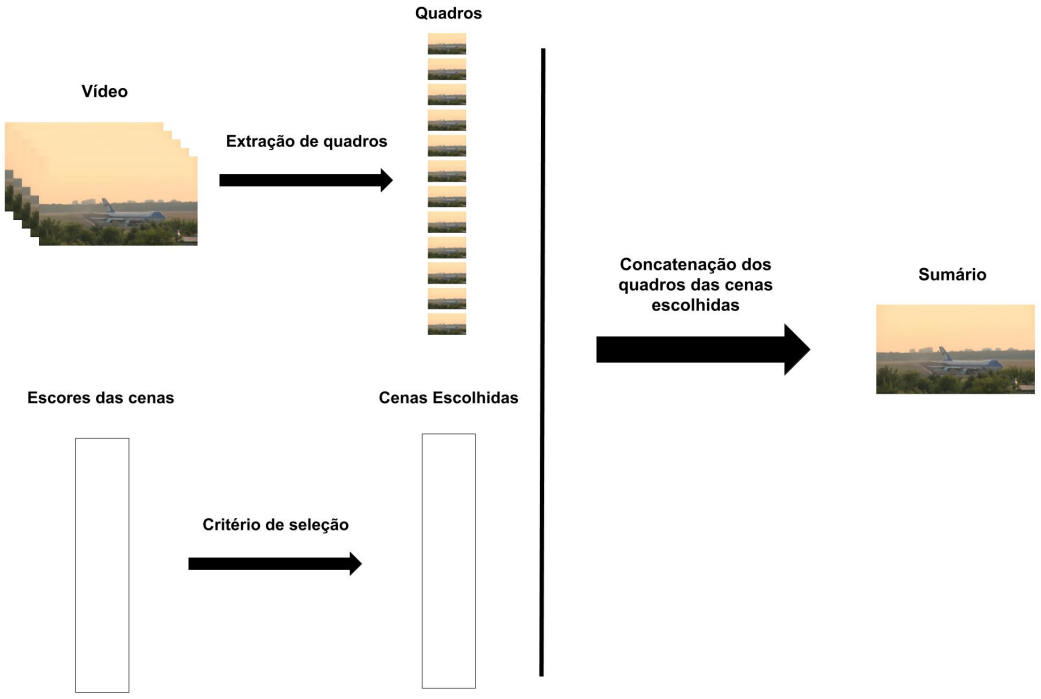

Figura 5.5: Diagrama geração de sumários assistíveis.

Nos trabalhos anteriores envolvendo sumarização, notou-se que o interesse maior era saber quais quadros/cenas devem compor o sumário, a fim de poder avaliar o modelo

usando alguma métrica (em trabalhos supervisionados, normalmente, usa-se o f-score). Porém, **poucos trabalhos geram, de fato, os sumários assistíveis**. Isso se deve ao foco principal desses: **Gerar modelos que apresentem uma maior acurácia nas bases de referência**. Neste trabalho, o interesse maior é **produzir estilos de sumário diferentes usando os critérios de seleção construídos**. Para isso, é importante poder, de fato, **assistir os sumários produzidos**.

Para possibilitar a geração de sumários assistíveis, extraiu-se todos os quadros dos vídeos da base SumMe e armazenou-os em formato *JPEG*. Em seguida, aplicou-se os estilos diferentes nos vídeos e identificou, para cada um, quais cenas deveriam aparecer em cada sumário. Com isso, realizou-se uma **concatenação das cenas** que compõem cada sumário, gerando, assim, sumários assistíveis. A Figura 5.5 apresenta esse processo com clareza.

5.5 Critérios de estilo propostos

Esta seção apresenta os critérios de seleção de cenas produzidos neste trabalho. O objetivo desses critérios é **produzir sumários com estilos diferentes**.

Os critérios de seleção produzidos foram:

- **Maior que a média, ou igual:** Neste critério de seleção, calcula-se a **média dos escores das cenas**. Em seguida, verifica-se quais cenas possuem um escore de seleção **maior ou igual** que o escore médio. Em caso positivo, marcam-se os quadros da cena com *1* (significando que esses fazem parte do sumário). Em caso negativo, marca-se os quadros da cena com *0* (logo, não fazem parte do sumário).
- **Menor que a média:** Como no caso anterior, começa-se calculando a média dos escores das cenas. Verifica-se quais cenas possuem um escore de seleção **menor** que o escore médio. Em caso positivo, marcam-se os quadros da cena com *1* (significando que esses fazem parte do sumário). Em caso negativo, marca-se os quadros da cena com *0* (logo, não fazem parte do sumário).
- **Maior que a média + 2.desvio padrão:** Neste caso, calcula-se a **média e o desvio padrão** das cenas. Os quadros que possuem um escore maior que a média somada com o dobro do desvio padrão entram para o sumário.
- **Randômico ponderado:** Diferente dos critérios anteriores, este critério seleciona quadros, ao invés de cenas. Cenas que apresentam o maior escore do vídeo não sofrem amostragem, aparecendo completas no sumário. Cenas com o menor escore do vídeo são removidas. Cenas com escore intermediário sofrem uma amostragem

de seus quadros inversamente proporcional ao escore de importância da cena (i.e., quanto maior o escore da cena, menos ela é amostrada).

Capítulo 6

Resultados Experimentais

Este capítulo apresenta os resultados experimentais obtidos na aplicação da metodologia proposta, evidenciando meios de gerar sumários diferentes a partir de vídeos, usando os escores de importância e critérios de seleção de cenas. Primeiramente, descreve-se os *softwares* utilizados na implementação da metodologia. Em seguida, explica-se como os experimentos foram realizados. Por fim, apresenta-se os experimentos realizados e analisa-se os resultados.

6.1 Softwares utilizados

Para a realização do trabalho, foram utilizados:

- Sistema Operacional *Ubuntu 18.04 LTS 64-bit* [46];
- Ambiente de computação interativa *Jupyter* [47];
- Linguagem de Programação *Python 2.7* [48], juntamente com as bibliotecas não-padrão *numpy* [49], para computação vetorial, *h5py* [50] para manipulação de dados em formato HDF5, *scipy* [51] para cálculos estatísticos e *matplotlib* [52] para visualização de dados.

6.2 Realização dos experimentos

Os experimentos foram realizados, para cada vídeo, da seguinte maneira:

- **Definição da estrutura do vídeo:** Nesta etapa, assistiu-se o vídeo e analisou sua estrutura, verificando os acontecimentos presentes nele;

- **Cálculo do escore de importância e comparação com estrutura:** Geração dos escores de importância a nível de quadro (apesar dos quadros em uma mesma cena possuírem o mesmo escore) e análise diante de sua estrutura;
- **Definição dos objetivos de sumarização:** Nesta etapa, estabelece-se quais características os sumários devem apresentar;
- **Criação do critério de seleção:** Implementa-se um critério de seleção de quadros, a fim de extrair, a partir dos objetivos e análises realizadas previamente, as cenas de acordo com as características estabelecidas;
- **Geração dos sumários e análise do resultado:** Por fim, gera-se o sumário assistível, e os gráficos de seleção, e analisa se os sumários produzidos condizem com os objetivos de sumarização pré-estabelecidos.

6.3 Experimento 1: Vídeo *Fire Domino*

6.3.1 Estrutura do vídeo

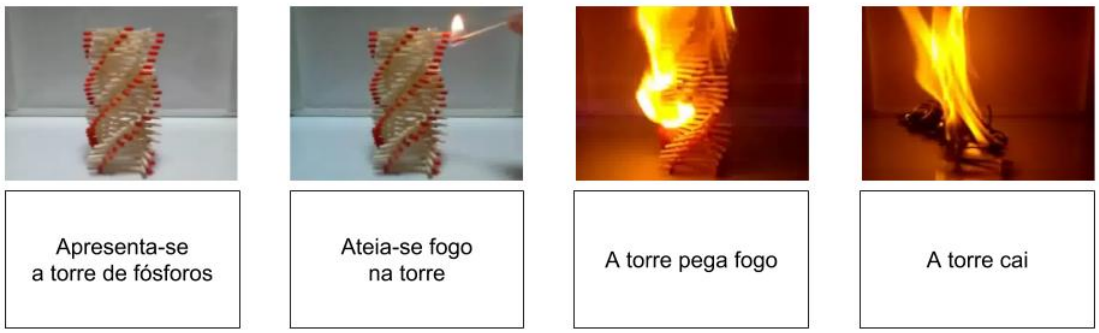

Figura 6.1: Estrutura do vídeo *Fire Domino*

A estrutura do vídeo *Fire Domino* é apresentada na Figura 6.1. Os acontecimentos são os que seguem:

- O vídeo inicia apresentando uma torre de fósforos;
- Momentos depois, ateia-se fogo na torre;
- A torre pega fogo por alguns momentos;
- A torre cai devido a degradação causada pelo fogo.

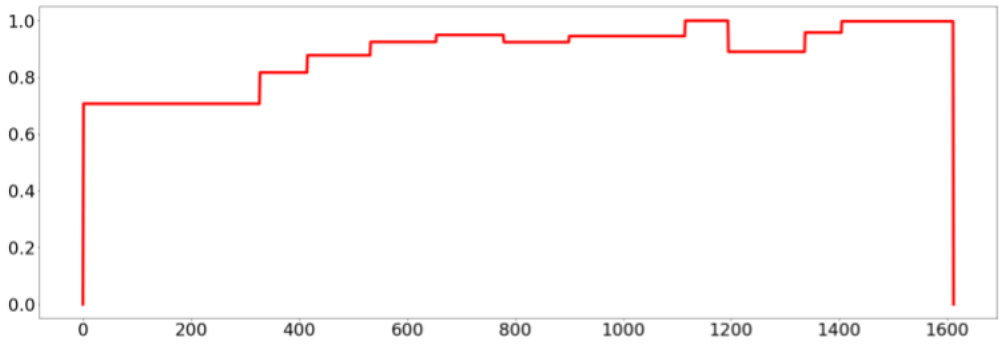

Figura 6.2: Importância dos quadros (Eixo Y) do vídeo *Fire Domino* ao longo do tempo (Eixo X)

6.3.2 Objetivos de sumarização

Para este vídeo, definiu-se dois objetivos: **Gerar um sumário que apresente os momentos antecedentes ao incêndio** e **gerar um sumário que apresente os acontecimentos durante o incêndio**.

Pensou-se nesses objetivos supondo uma situação de **análise de gravações de incêndio reais**. Nessa, as cenas contendo os momentos em que o incêndio já está instalado podem apresentar maior **escore de importância**, porém, **cenas prévias ao incêndio são importantes para a verificação do que o causou**.

6.3.3 Seleção de cenas e análise dos sumários

A Figura 6.2 apresenta as importâncias dos quadros do vídeo. Foi possível observar que os quadros iniciais apresentam menor importância e os quadros seguintes tendem a progredir em importância até o fim do vídeo. Neste contexto, para obter os sumários definidos nos objetivos de sumarização, utilizou-se dois estilos de sumarização: O estilo **Maior que a média, ou igual** e o estilo **Menor que a média**.

As Figuras 6.4 e 6.3 apresentam de maneira gráfica quais cenas foram escolhidas para compor cada um dos sumários. Quadros em vermelho são aqueles que foram selecionados para compor o sumário, enquanto que os outros foram removidos.

No sumário gerado pelo estilo *Menor que a média*, obteve-se os momentos iniciais do vídeo, na qual apresenta-se a torre de fósforos, e o início do incêndio. Com esse estilo, **geramos um sumário que apresenta os momentos que antecedem o incêndio e o momento que apresenta o que o causou**. Usando o estilo *Maior que a média ou igual*, obtemos um sumário que apresenta **as cenas intermediárias e finais do incêndio e a queda da torre**.

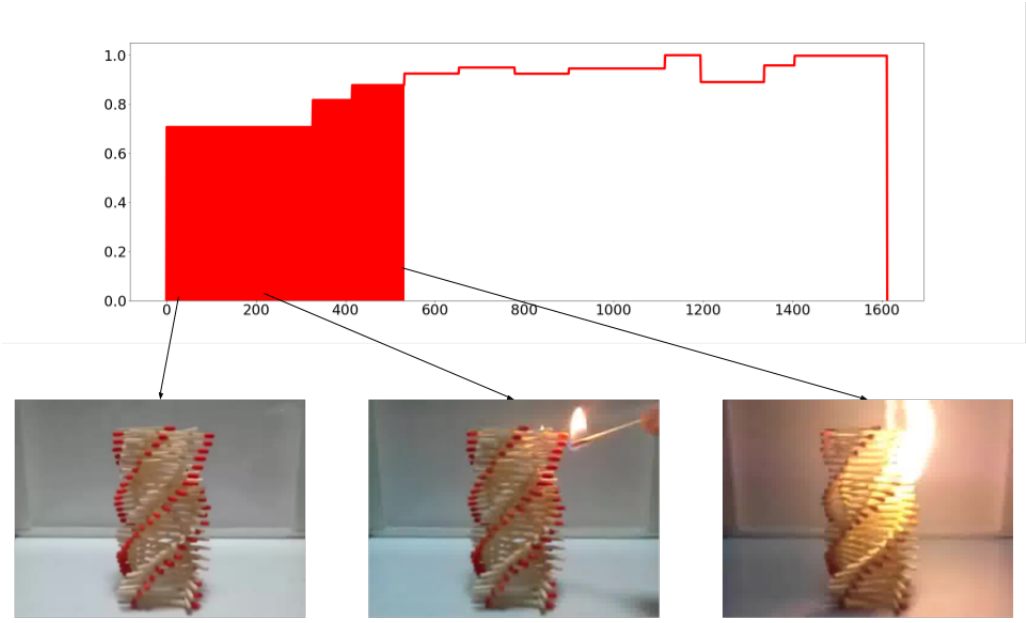

Figura 6.3: Cenas escolhidas aplicando o estilo *Menor que a média* em *Fire Domino*

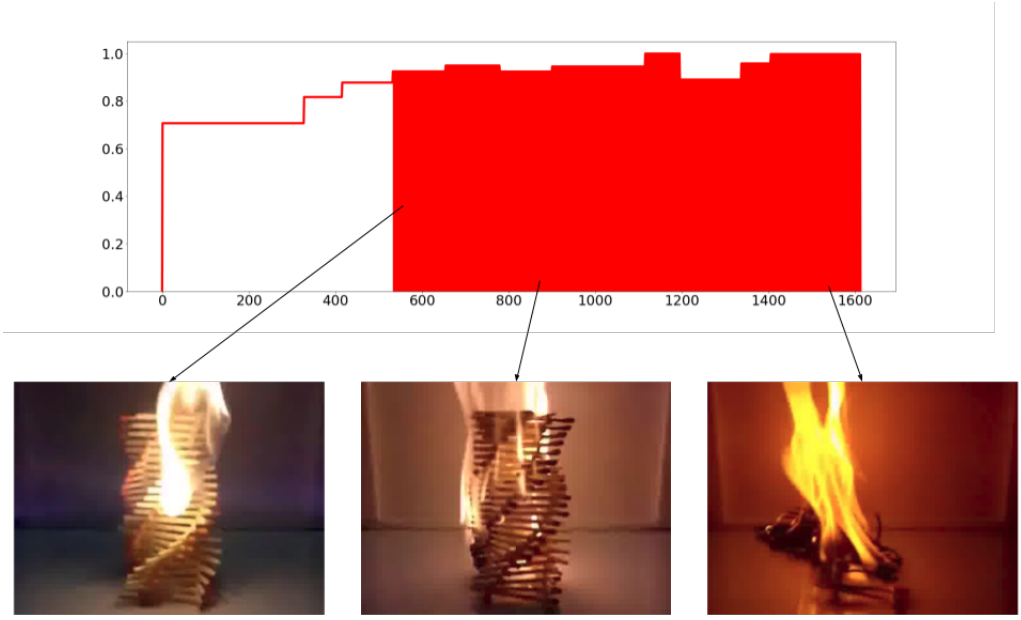

Figura 6.4: Cenas escolhidas aplicando o estilo *Maior que a média ou igual* em *Fire Domino*

6.4 Experimento 2: Vídeo *Uncut Evening Flight*

6.4.1 Estrutura do vídeo

Figura 6.5: Estrutura do vídeo *Uncut Evening Flight*

Esse vídeo apresenta a seguinte estrutura: No início, o avião de controle remoto decola. As cenas intermediárias do vídeo são *panoramas* do avião e do cenário no fundo. Como a câmera foi acoplada na asa do avião, **diversos cenários aparecem, enquanto que o avião permanece estático ao longo do vídeo**. Por fim, o avião pousa.

6.4.2 Objetivos de sumarização

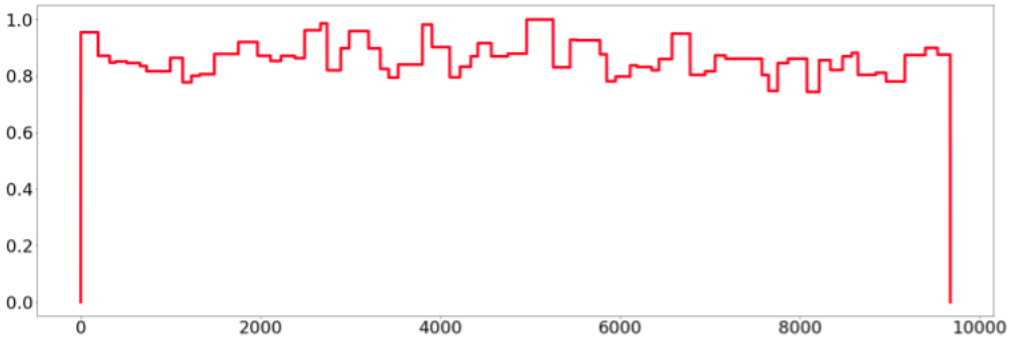

Figura 6.6: Importância dos quadros do vídeo *Uncut Evening Flight*

O vídeo apresenta uma característica interessante: Apresenta-se várias cenas em que **o avião permanece estático na gravação, enquanto que o fundo muda de acordo**

com o vôo. Com isso, decidiu-se pelo seguinte objetivo de sumarização: **Apresentar as cenas intermediárias em cortes, de maneira que o avião permaneça estático e o cenário no fundo mude repentinamente**. Isso é muito realizado em edições de vídeo, a fim de acrescentar estética.

Na Figura 6.6, foi possível notar que os escores de maior importância são os da decolagem, no início do vídeo, os do pouso, no final, e algumas das cenas intermediárias, as quais apresentam panoramas. Além disso, **os escores das cenas panorâmicas são maiores que os escores das cenas de pouso e decolagem**.

6.4.3 Seleção de cenas e análise dos sumários

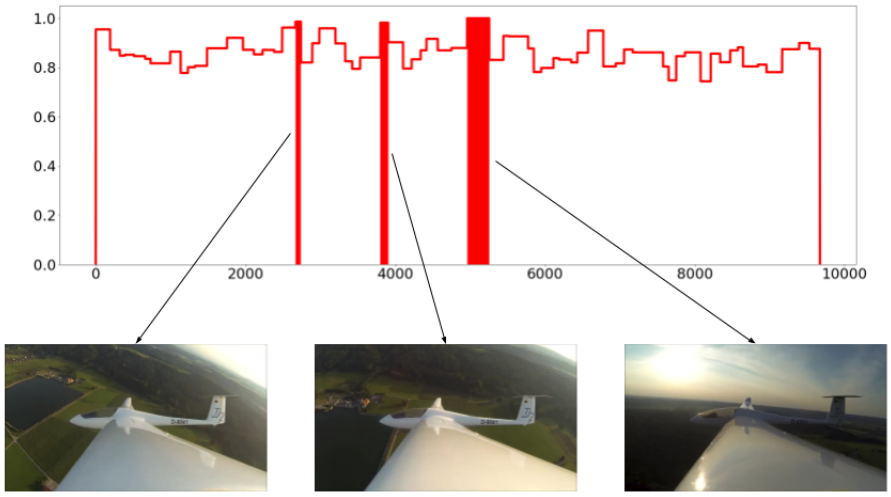

Figura 6.7: Cenas escolhidas aplicando o estilo *Maior que a média+2.desvio padrão* em *Uncut Evening Flight*

Sabendo que os escores altos das cenas intermediárias são os maiores do vídeo, utilizou-se o estilo **Maior que a média + 2.desvio padrão**. Com isso, somente os escores mais altos do vídeo foram selecionados. Além disso, ao concatenar as cenas, o efeito de transição em corte desejado foi obtido.

6.5 Experimento 3: Vídeo *Base jumping*

6.5.1 Estrutura do vídeo

O diagrama da Figura 6.8 apresenta a estrutura do vídeo *Base jumping*. Esse apresenta uma prática de *base jumping* com *wingsuit* [53] com gravação em primeira pessoa (câmera

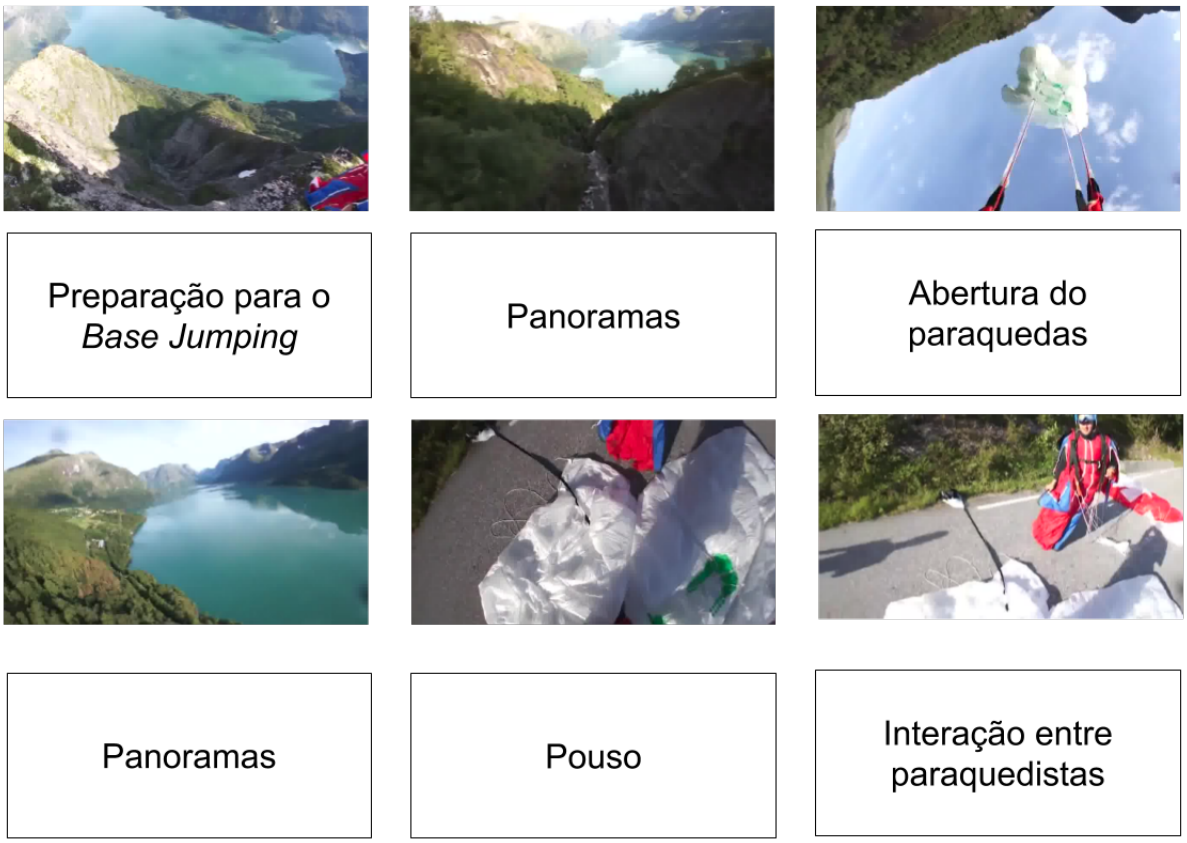

Figura 6.8: Estrutura do vídeo *Base jumping*

acoplada no capacete do desportista). O vídeo começa com a preparação para o salto, mostrando a queda livre. Em seguida, durante a queda, cenas panorâmicas do cenário são apresentadas. Depois, uma mudança brusca ocorre no vídeo, no momento da abertura do paraquedas, sendo esse o foco da gravação no momento. Após a abertura do paraquedas, mais cenas panorâmicas mostram o cenário das montanhas acompanhadas do rio. Por fim, o desportista pousa e interage com outro paraquedista.

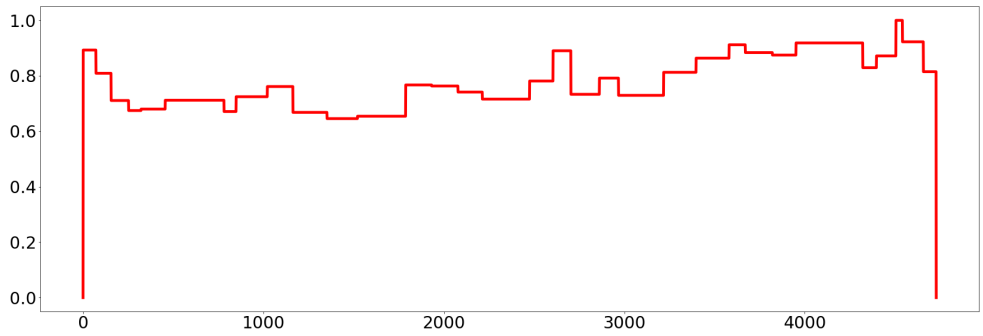

Figura 6.9: Importância dos quadros do vídeo *Base jumping*

6.5.2 Objetivos de sumarização

Observe que esse vídeo apresenta uma grande variedade de informação. As cenas de preparação para o pulo, abertura do paraquedas, pouso e interação com paraquedista são **todas relacionadas ao esporte**. Em contraste, as cenas da queda livre e das gravações entre a abertura do paraquedas e o pouso **apresentam foco nas paisagens**. Neste contexto, foram estabelecidos dois objetivos de sumarização: Extrair, em um sumário, as partes relativas ao **paraquedismo**, enquanto que, em outro sumário, apresentar as **paisagens** gravadas durante o salto.

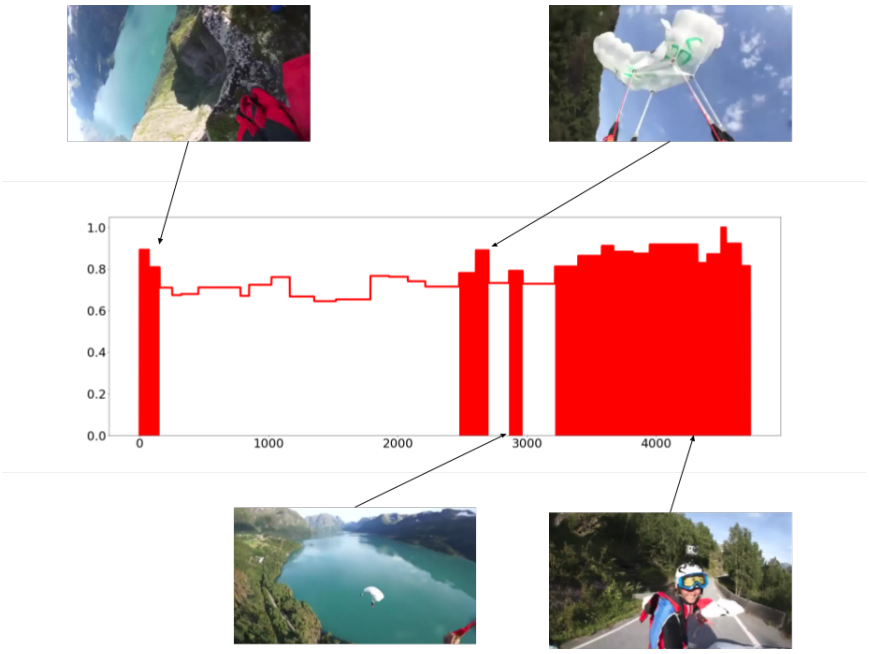

Figura 6.10: Cenas escolhidas aplicando o estilo *Maior que a média ou igual* em *Base jumping*

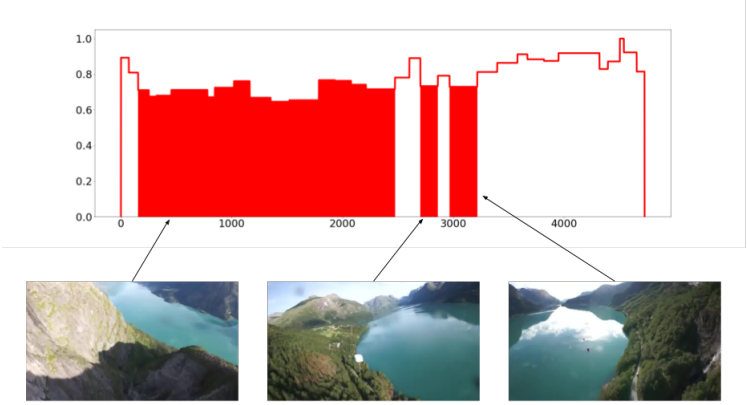

Figura 6.11: Cenas escolhidas aplicando o estilo *Menor que a média* em *Base jumping*

6.5.3 Seleção de cenas e análise dos sumários

A Figura 6.9 apresenta os escores de importância normalizados dos quadros. Foi possível perceber que as cenas relativas ao paraquedismo apresentaram um **maior escore de importância**, enquanto que as cenas das paisagens apresentaram **um escore de importância reduzido**. Com isso, usou-se os mesmos estilos aplicados no vídeo *Fire Domino* para obter os sumários desejados: *Maior que a média ou igual* e *Menor que a média*.

O sumário gerado pelo estilo *Maior que a média ou igual* apresenta, majoritariamente, as cenas de preparação para o *base jumping*, a abertura do paraquedas, o pouso e a interação entre os paraquedistas, ou seja, as cenas relacionadas ao paraquedismo. Enquanto que o sumário gerado pelo estilo *Menor que a média* incorporou as cenas cujo foco é nas paisagens.

6.6 Experimento 4: Vídeo *Valparaiso Downhill*

6.6.1 Estrutura do vídeo

A Figura 6.12 apresenta o vídeo, em primeira pessoa, apresentando uma prática de *urban mountain biking* [54], esporte em que se realiza trilhas usando uma *mountain bike* em espaços urbanos. Começa-se com a preparação do desportista e a partida. Em seguida, apresenta diversas cenas da descida do *Valparaiso Downhill*, travessia de obstáculos e manobras. Por fim, chega-se no final da descida, em que o desportista é recebido por espectadores.

Figura 6.12: Estrutura do vídeo *Valparaiso Downhill*

6.6.2 Objetivos de sumarização

Esse vídeo, apesar da variedade de situações, apresenta uma **constância**, no sentido de que não ocorrem muitas cenas que sejam mais relevantes que outras. Isso pode ser

observado nas importâncias dos quadros do vídeo, apresentadas na Figura 6.13. Note que **a maioria das cenas apresentam uma importância similar**, entre 0.7 e 0.9. Com isso, decidiu-se pelo seguinte objetivo de sumarização: Gerar um sumário que contenha **toda a informação sequencial do vídeo, mas que seja mais rápido de assistir**.

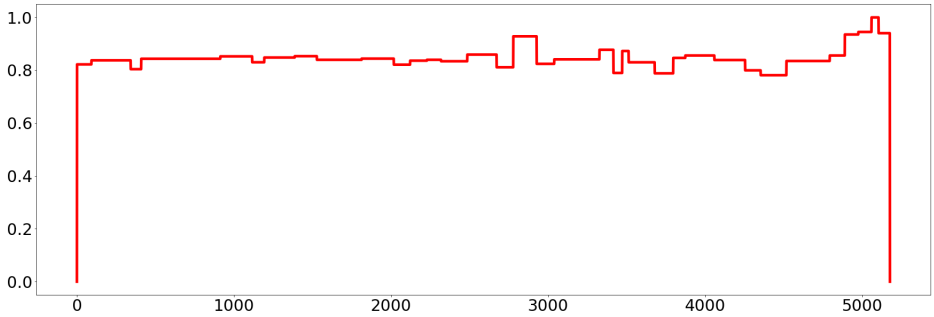

Figura 6.13: Importâncias dos quadros em *Valparaiso Downhill*

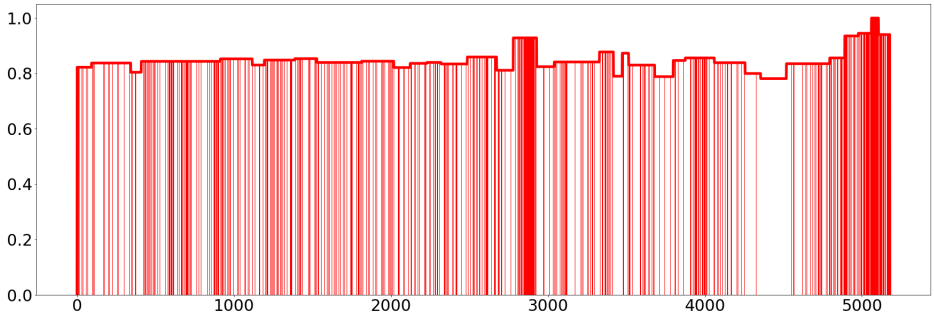

Figura 6.14: Quadros escolhidos aplicando o estilo *Randômico ponderado* em *Valparaiso Downhill*

6.6.3 Seleção de cenas e análise do sumário

Uma forma de realizar o objetivo de sumarização proposto é utilizando o estilo *Randômico ponderado por escore*, o qual é um **critério de seleção de quadros**. Diferente dos outros estilos usados até agora, esse seleciona **quadros**, ao invés de **cenas**, ou seja, desconsidera a segmentação do vídeo.

Como pode ser observado na Figura 6.14, os quadros que compõem o sumário apresentaram maior chance de ser selecionado caso seu escore de importância seja mais elevado. No caso, cenas cujo escore é máximo têm 100% de chance de aparecer no sumário, isso faz com que essas cenas passem em velocidade normal, cenas com o menor escore do vídeo

não aparecem e cenas intermediárias sofrem uma amostragem dos seus quadros, sendo menos amostrados quanto maior seus escores. Com isso, o sumário apresentado é mais rápido de assistir, pois passa mais rápido nas partes menos importantes, e foca nas cenas mais importantes.

6.7 Experimento 5: Aplicação dos estilos em 5 vídeos diferentes

Nesta seção, demonstra-se a aplicação de três dos critérios de seleção discutidos até o momento, *Maior que a média ou igual*, *Menor que a média* e *Randômico ponderado*, em 5 outros vídeos da base de dados SumMe: *Airforce one*, *Bearpark climbing*, *Bus in rock tunnel*, *Car railcrossing* e *Jumps*. Exemplos de quadros desses são apresentados na Figura 6.15.

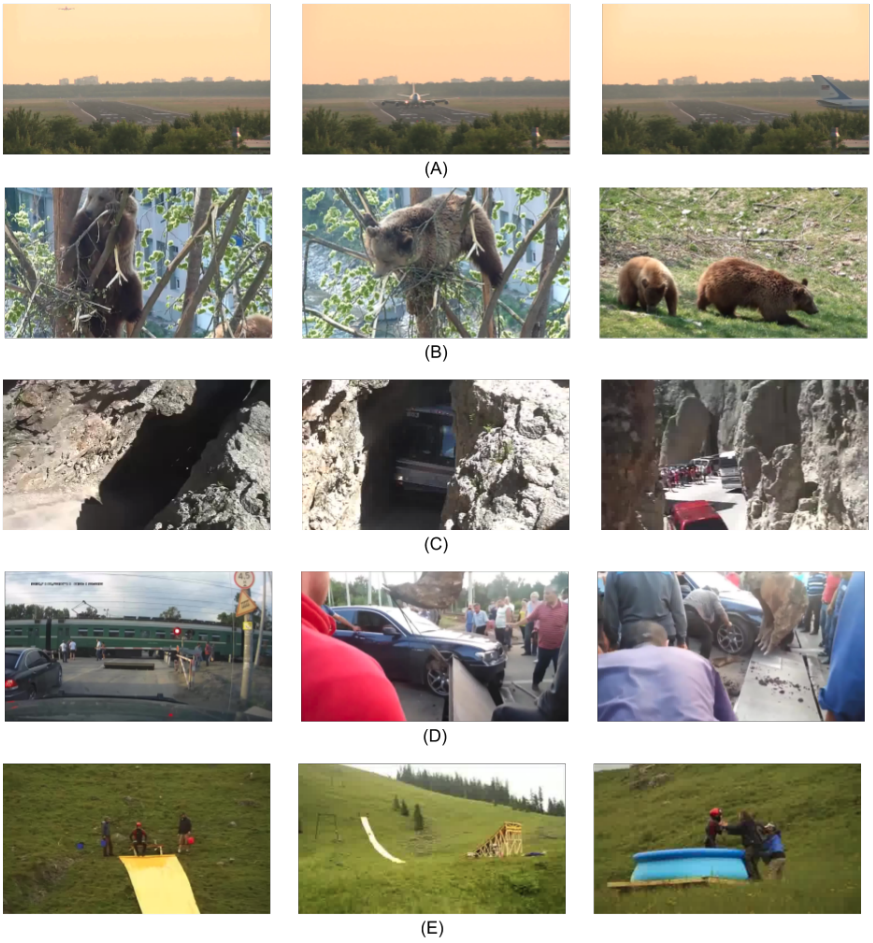

Figura 6.15: Amostra sequencial de 3 quadros presentes em (A) *Airforce one*, (B) *Bearpark climbing*, (C) *Bus in rock tunnel*, (D) *Car railcrossing* e (E) *Jumps*.

O vídeo *Airforce one* apresenta, em terceira pessoa, a gravação estática do pouso de um avião, começando com a gravação da pista, seguida da aproximação do avião. Com isso, o avião pousa e, na medida em que segue na pista de pouso, se aproxima da câmera. Por fim, o avião faz uma curva e desaparece gradativamente da gravação. Em *Bearpark climbing*, ursos interagem e sobem em árvores ao longo de todo o vídeo. *Bus in rock tunnel* apresenta uma gravação de um ônibus passando com dificuldade por um túnel de pedra estreito. *Car railcrossing* é uma gravação de um carro preso em uma barreira de trilho de trem, e as tentativas de retirá-lo. Por fim, o vídeo *Jumps* mostra uma pessoa escorregando em uma rampa, sendo lançada no ar e aterrissando em uma piscina rasa.

As Figuras 6.16 à 6.30 mostram, de maneira gráfica, a seleção de quadros/cenas, por estilo, para cada vídeo.

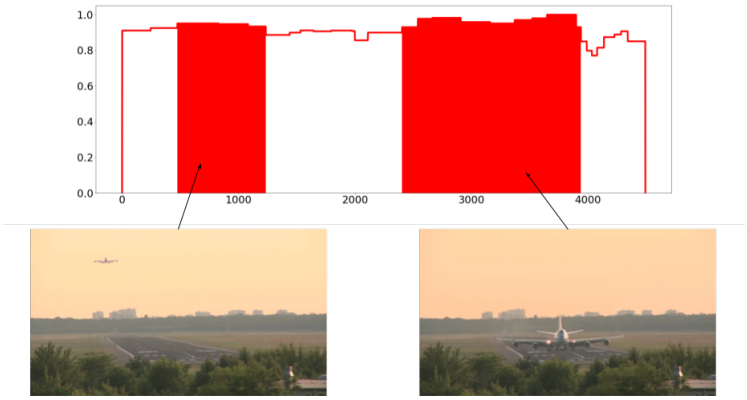

Figura 6.16: Sumarização usando o estilo *Maior que a média ou igual* no vídeo *Airforce one*

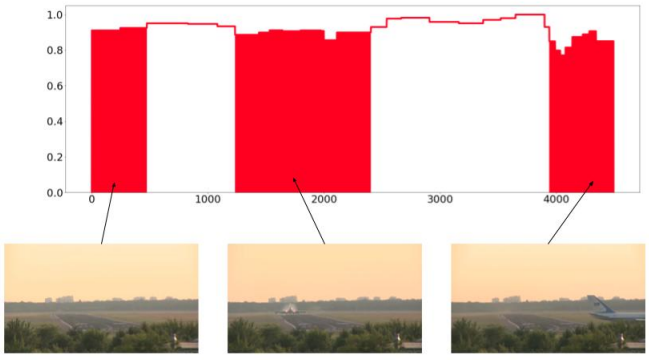

Figura 6.17: Sumarização usando o estilo *Menor que a média* no vídeo *Airforce one*

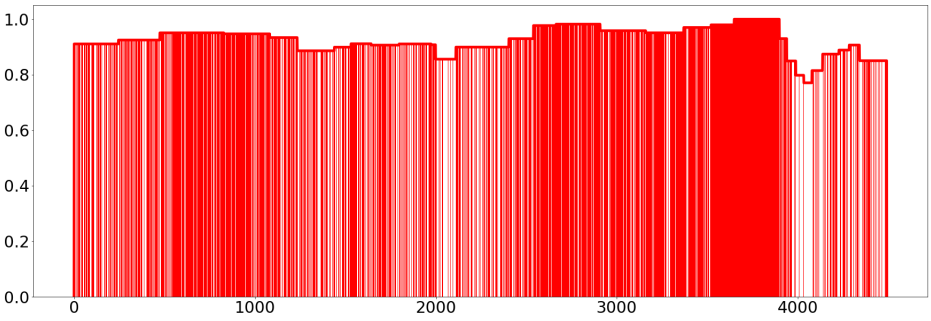

Figura 6.18: Sumarização usando o estilo *Randômico Ponderado* no vídeo *Airforce one*

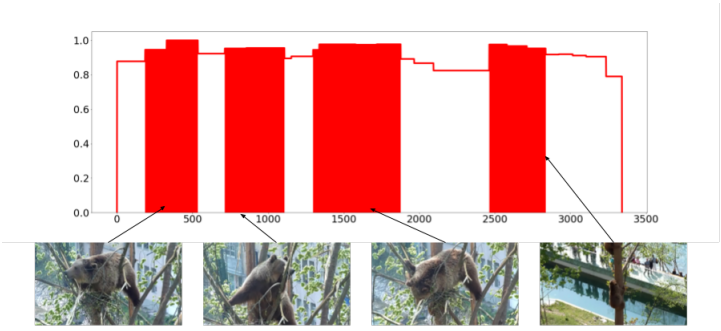

Figura 6.19: Sumarização usando o estilo *Maior que a média ou igual* no vídeo *Bearpark climbing*

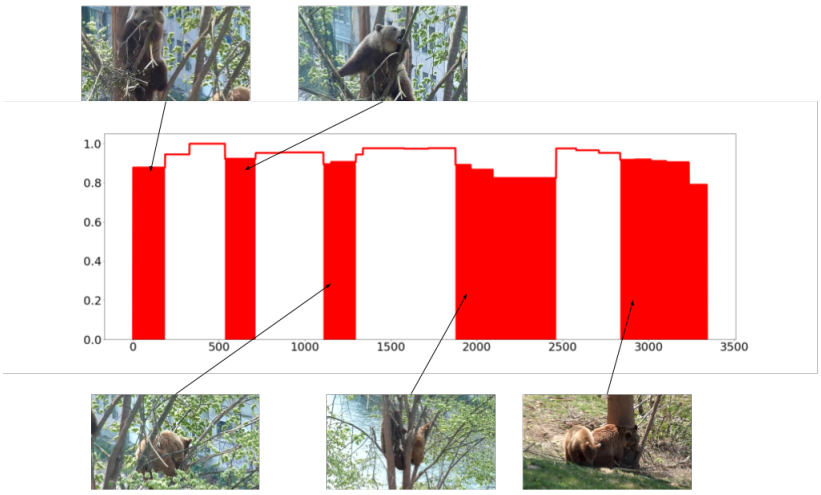

Figura 6.20: Sumarização usando o estilo *Menor que a média* no vídeo *Bearpark climbing*

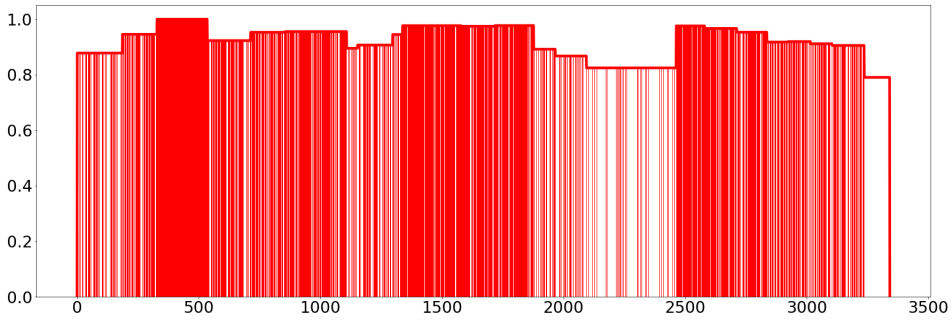

Figura 6.21: Sumarização usando o estilo *Randômico Ponderado* no vídeo *Bearpark climbing*

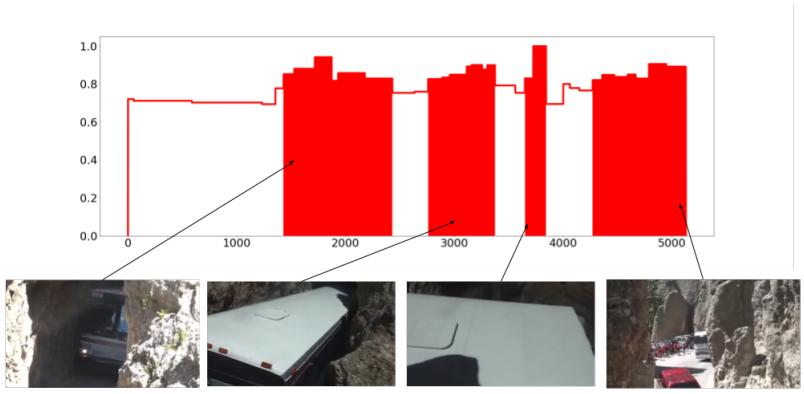

Figura 6.22: Sumarização usando o estilo *Maior que a média ou igual* no vídeo *Bus in rock tunnel*

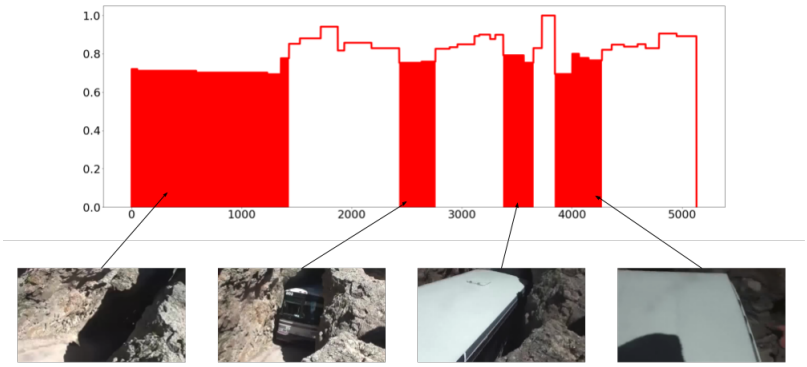

Figura 6.23: Sumarização usando o estilo *Menor que a média* no vídeo *Bus in rock tunnel*

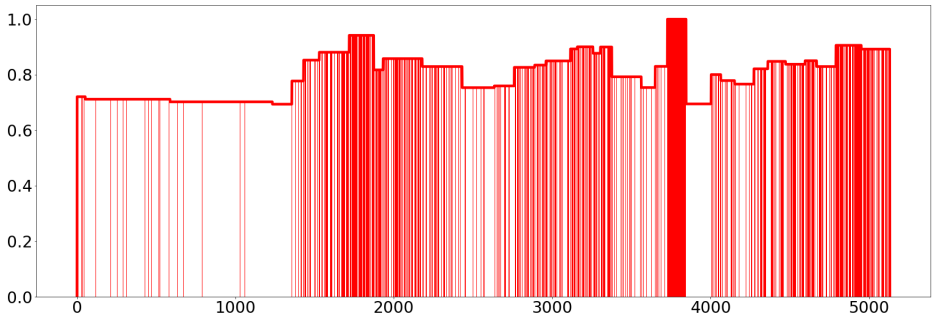

Figura 6.24: Sumarização usando o estilo *Randômico Ponderado* no vídeo *Bus in rock tunnel*

Figura 6.25: Sumarização usando o estilo *Maior que a média ou igual* no vídeo *Car railcrossing*

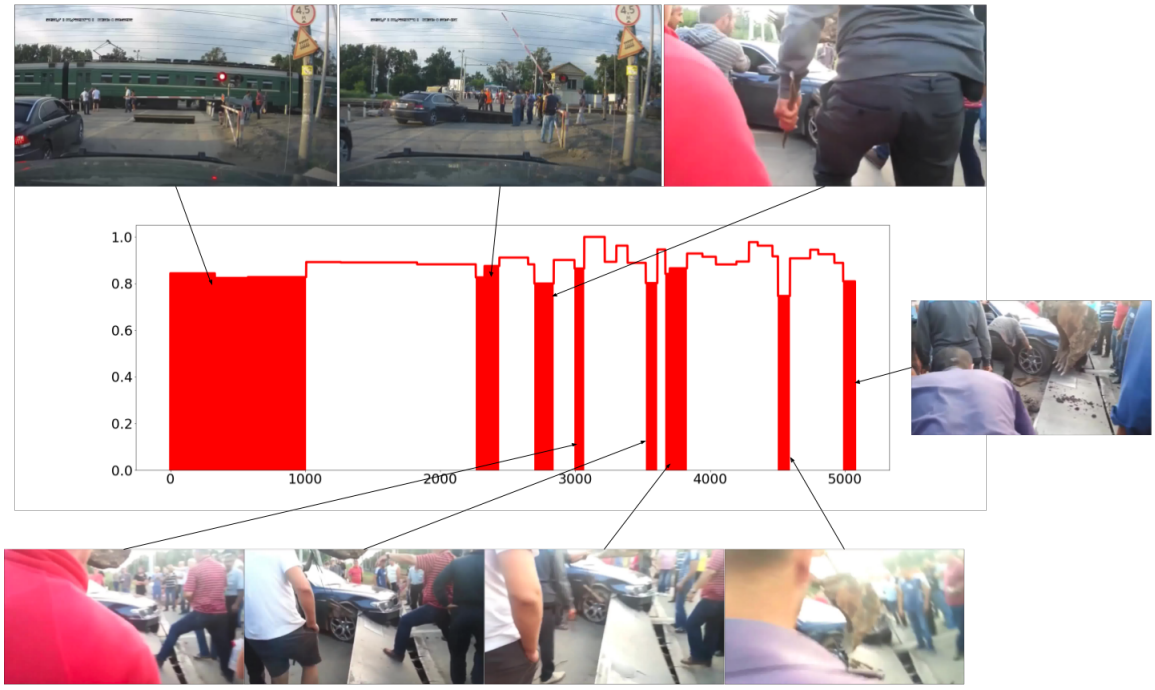

Figura 6.26: Sumarização usando o estilo *Menor que a média* no vídeo *Car railcrossing*

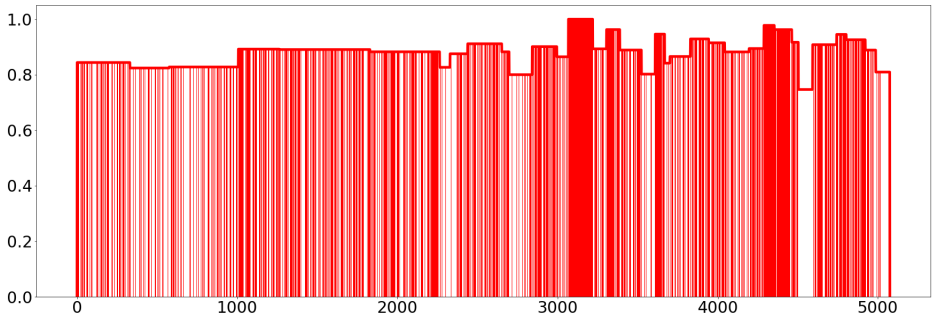

Figura 6.27: Sumarização usando o estilo *Randômico Ponderado* no vídeo *Car railcrossing*

Figura 6.28: Sumarização usando o estilo *Maior que a média ou igual* no vídeo *Jumps*

Figura 6.29: Sumarização usando o estilo *Menor que a média* no vídeo *Jumps*

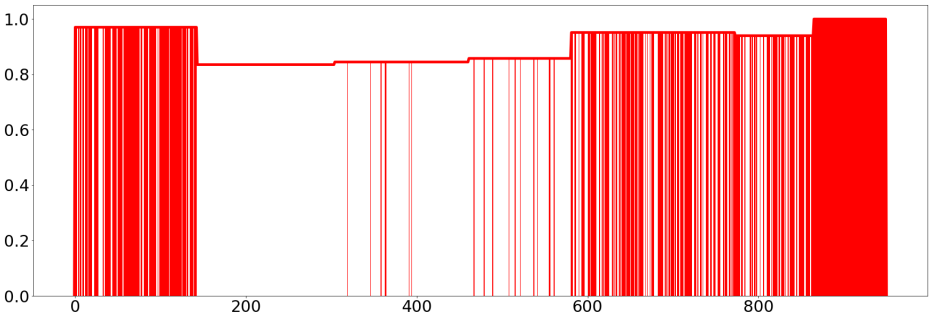

Figura 6.30: Sumarização usando o estilo *Randômico Ponderado* no vídeo *Jumps*

6.7.1 Análise dos sumários gerados

Esta subseção analisa, de maneira geral, todos os sumários produzidos no experimento 5, apresentando as características de cada um.

Vídeo: *Airforce One*

- *Estilo Maior que a média ou igual:* O sumário começa com o avião sobrevoando a pista de pouso e preparando para pousar. Em seguida, mostra-se o avião percorrendo parte da pista de pouso;
- *Estilo Menor que a média:* Apresenta-se as cenas iniciais, em que o avião ainda não aparece no cenário, o início do pouso e a cena em que o avião faz a curva no final;
- *Estilo Randômico Ponderado:* O sumário apresenta todo o vídeo, porém acelerado. A parte do vídeo em que o avião está prestes a virar é a que apresenta mais quadros, sendo, assim, em velocidade normal. Como a cena da curva do avião é a com menor escore, é praticamente excluida do sumário.

O sumário *Maior que a média ou igual* do vídeo *Airforce one* é ideal para **saber a história do vídeo**. Assistindo ele, sabe-se imediatamente que é um vídeo sobre pouso de avião. Já o sumário *Menor que a média* é bom para **analisar as partes críticas do vídeo**: A cena exata do pouso e da curva do avião. Por fim, o sumário *Randômico Ponderado* apresenta praticamente toda a informação do vídeo, porém de maneira acelerada. Com ele, sabe-se exatamente o que aconteceu no vídeo em detalhe, porém, em menor tempo.

Vídeo: *Bearpark climbing*

- *Estilo Maior que a média ou igual:* Esse sumário apresenta, inicialmente, o primeiro urso da gravação escalando uma árvore. Em seguida, mostra-se o mesmo urso descendo a árvore, acompanhado de um filhote;
- *Estilo Menor que a média:* Começa de maneira similar ao sumário anterior, mostrando o primeiro urso da gravação. Porém, nas cenas intermediárias e finais, foca em outros ursos e na interação desses entre si;
- *Estilo Randômico Ponderado:* O sumário gerado por esse estilo apresenta todas as partes do vídeo original. As cenas em que se mostra o primeiro urso escalando são as que passam em velocidade normal, enquanto que a cena final (primeiro urso e filhote caminhando juntos) é excluída do sumário.

Analisando os sumários de *Bearpark climbing*, verificou-se que o sumário *Maior que a média ou igual* possui um **enfoque exclusivo no primeiro urso** que aparece na gravação. Já o sumário *Menor que a média* apresenta mais cenas da **interação entre os ursos, focando, também, na escalada dos outros**. Já o sumário *Randômico ponderado* é o melhor para assistir **todas as cenas de escalada**, pois esse exclui parcialmente os quadros em que os ursos estão no chão.

Vídeo: *Bus in rock tunnel*

- *Estilo Maior que a média ou igual:* Apresenta-se a cena em que o ônibus já surgiu do túnel escuro. Em seguida, mostra-se o ônibus na metade do caminho de saída do túnel. O sumário finaliza com a saída completa do ônibus, o qual vai de encontro com outros carros estacionados;
- *Estilo Menor que a média:* Esse sumário começa com o túnel escuro, mostrando apenas o farol do ônibus. Em seguida, mostra-se o ônibus passando pelo trecho mais estreito. Por fim, o ônibus finalizando a passagem;
- *Estilo Randômico Ponderado:* Esse sumário é uma versão rápida do vídeo original. As partes em que o ônibus já surgiu do túnel, passou pela metade do túnel e vai de encontro com os carros são mostradas em velocidade menor, enquanto que as cenas iniciais, em que o ônibus ainda não surgiu do túnel e a cena em que o ônibus passa pelo trecho estreito são aceleradas. Grande parte da cena inicial é removida.

O sumário *Maior que a média ou igual* apresenta todas as **informações relevantes do vídeo**, sendo possível, por meio dele, entender todos os acontecimentos apresentados no vídeo original. Já o sumário *Menor que a média* apresenta trechos de menor importância para o entendimento do vídeo original, porém, são cenas interessantes. A cena inicial, em que o ônibus não surgiu do túnel, adiciona curiosidade ao sumário. A cena em que o ônibus passa pela parte mais estreita adiciona tensão, pois dá a sensação de que o ônibus não será capaz de passar. Logo, o estilo *Menor que a média* **adiciona um aspecto estético** ao vídeo original. Por fim, o *Randômico Ponderado* conta **toda a história presente no sumário** *Maior que a média ou igual*, **acelerando as outras cenas** e excluindo parcialmente as cenas presentes no sumário *Menor que a média*.

Vídeo: *Car railcrossing*

- *Estilo Maior que a média ou igual:* O sumário começa com o carro já enguiçado na barreira dos trilhos. Em seguida, mostra-se as cenas em que um trator tenta remover o carro;

- *Estilo Menor que a média:* Esse sumário apresenta a cena exata em que o carro fica preso e momentos antes do acontecimento. A parte final mostra cenas em que a gravação ficou prejudicada devido ao trânsito de pessoas na frente.
- *Estilo Randômico Ponderado:* Apresenta o vídeo original acelerado. As cenas em que o trator tenta remover o carro são as que passam em velocidade normal, enquanto que as cenas do trânsito de pessoas são as que não aparecem.

Após a análise dos sumários desse vídeo, decidiu-se que o sumário *Maior que a média ou igual* é o melhor para, inicialmente, entender o problema (carro preso) e assistir, em seguida, as cenas da tentativa de remoção do carro, **removidas as cenas do trânsito de pessoas**. O sumário *Menor que a média* é o melhor para se entender **como o carro ficou preso**, porém as cenas do trânsito de pessoas **não acrescentam informação**, sendo um defeito do sumário. *Randômico Ponderado* apresenta a história completa, **excluindo parcialmente as cenas de movimentação das pessoas em frente a câmera**.

Vídeo: *Jumps*

- *Estilo Maior que a média ou igual:* Esse sumário apresenta as cenas iniciais do vídeo, as quais mostram a preparação da descida da rampa. Por fim, mostra-se as cenas pós salto, onde a pessoa que o performou já está na piscina de plástico;
- *Estilo Menor que a média:* Esse estilo selecionou uma parte contínua do vídeo, onde a pessoa desce a rampa, é lançada no ar e, por fim, cai na piscina de plástico;
- *Estilo Randômico Ponderado:* Praticamente igual ao sumário *Maior que a média ou igual*, porém mais acelerado. As cenas que compõem o sumário *Menor que a média* foram, praticamente, totalmente excluídas.

No caso desse vídeo, o sumário gerado pelo estilo *Randômico Ponderado* é confuso, já que **passa muito rápido**. O estilo *Maior que a média ou igual* também apresenta um grau de confusão, pois só mostra as cenas pré e pós lançamento, **não apresentando o que, de fato, aconteceu entre essas**. Por fim, o sumário *Menor que a média* foi o mais interessante, uma vez que apresenta **a descida, lançamento e mergulho na piscina**.

6.8 Análise dos resultados

A partir dos experimentos realizados, observou-se que é possível definir um processo de sumarização baseado em objetivos específicos, a partir dos escores de importância de cenas (ou quadros). Ou seja, consegue-se, a partir da análise dos escores de importância, do

contexto do vídeo e da necessidade de sumarização, definir um critério de seleção capaz de gerar um sumário com as características desejadas.

Isso foi demonstrado nos experimentos, onde, tendo o contexto do vídeo, uma análise gráfica dos escores de importância e objetivos de sumarização específicos, foi possível estabelecer critérios de seleção de cenas capazes de gerar os sumários esperados.

Uma característica importante sobre os critérios de seleção é a simplicidade desses. Todos os critérios usam conceitos básicos de matemática e estatística e, com isso, são capazes de produzir sumários interessantes e funcionam em diversos contextos diferentes (i.e. a aplicação dos estilos *Maior que a média ou igual* e *Menor que a média* nos vídeos *Fire Domino* e *Base jumping*, os quais apresentaram situações diferentes).

Outro aspecto relevante dos resultados foi o maneira com que foi realizada a sumarização no experimento 4. Até onde foi verificado, nenhum outro trabalho anterior sobre sumarização automática de vídeo abordou um método similar de sumarização, onde se realiza uma **amostragem proporcional à importância do quadro no vídeo**. Dessa forma, este trabalho apresenta uma maneira nova de se sumarizar vídeos automaticamente.

Capítulo 7

Conclusões

Neste trabalho, demonstrou-se que é possível gerar sumários com estilos diferentes, usando os escores de importância, gerados por um modelo supervisionado previamente treinado, de quadros e cenas de um vídeo, a partir da criação de critérios de seleção baseados nos escores de importância.

Primeiramente, na Seção 5.1, foi introduzido **o conceito de estilo de sumarização** em vídeos. Com isso, foram criados critérios de seleção de cenas, baseados nos escores de importância gerados pelo modelo dppLSTM, a fim de demonstrar a criação de sumários com características visuais e sequenciais diferentes entre si. No Capítulo 6, experimentos foram realizados analisando a estrutura de alguns vídeos componentes da base SumMe, calculou-se os escores de importância desses usando o modelo dppLSTM pré treinado usando as bases OVP, Youtube e TVSUM, foram verificadas as relações entre escores de importância e estrutura do vídeo, estabelecendo características desejadas para sumários dos vídeos, gerando critérios de seleção, a fim de obter os sumários estabelecidos e, por fim, verificou se as características dos sumários gerados correspondem com as características desejadas. Além disso, foram aplicados os critérios produzidos nos experimentos passados em mais 5 vídeos e analisadas as características presentes nos sumários produzidos, evidenciando pontos fortes, fracos e situações em que um estilo é preferível em relação a outro.

A principal contribuição deste trabalho é a demonstração de que existem várias formas de se sumarizar, sendo que cada maneira apresenta aspectos diferentes do vídeo original. Com isso, acredita-se que **a melhor maneira de se informar usando sumários é, primeiramente, estabelecer o que se deseja obter do vídeo original**. Durante o estudo de outras formas de sumarizar, usando o estilo *Randômico Ponderado*, criou-se um método de sumarização diferente dos aplicados nos trabalhos relacionados, possibilitando o estudo de novas abordagens de sumarização. Por fim, com o que foi apresentado no manuscrito, consideramos que os objetivos propostos foram alcançados.

7.1 Trabalhos Futuros

Nos experimentos apresentados no Capítulo 5, nota-se que os critérios de seleção criados foram desenvolvidos sob medida para cada vídeo e para cada objetivo de sumarização. Neste contexto, percebeu-se que alguns estilos foram mais genéricos que outros, sendo utilizado para mais de um vídeo e para mais de um objetivo de sumarização. Com isso, um trabalho futuro seria criar um **critério de seleção paramétrico**, a partir do qual, por meio apenas do ajuste de parâmetros no estilo, fosse possível atender uma maior diversidade de contextos de sumarização.

Na proposta deste trabalho, a sumarização ocorre em duas etapas: *(i)* Aplicação dos vídeos no modelo supervisionado, realizando o cálculo dos escores e *(ii)* seleção dos quadros/cenas que irão compor o vídeo. Dessa maneira, percebe-se que há uma separação entre as etapas. Portanto, como outra possibilidade de trabalho futuro, deseja-se produzir um modelo supervisionado capaz de, em sua saída, produzir a versão final do sumário. Ou seja, o modelo seria capaz de aplicar estilo diretamente, não sendo necessário a aplicação de um critério de seleção manualmente definido. Neste contexto, produziria-se um modelo genérico, o qual fosse possível refinar, a fim de atender uma necessidade de sumarização específica.

Referências

[1] *Web of Science*. https://www.webofknowledge.com. Último acesso: 2019-02-15.

[2] Goodfellow, Ian, Yoshua Bengio e Aaron Courville: *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[3] *Exemplo de CNN*. http://rafaelsakurai.github.io/cnn-mapreduce/. Último acesso: 2019-01-28.

[4] *Recurrent Neural Networks and LSTM*. https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5. Último acesso: 2019-02-11.

[5] Zhang, Ke, Wei Lun Chao, Fei Sha e Kristen Grauman: *Video summarization with long short-term memory*. Em *ECCV*. Springer, 2016.

[6] Lu, Zheng e Kristen Grauman: *Story-driven summarization for egocentric video*. Em *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[7] *Youtube*. https://www.youtube.com. Último acesso: 2019-01-16.

[8] Zhang, Hong Jiang, Jianhua Wu, Di Zhong e Stephen W. Smoliar: *An integrated system for content-based video retrieval and browsing*. Pattern Recognition, 30(4):643 – 658, 1997, ISSN 0031-3203. http://www.sciencedirect.com/science/article/pii/S0031320396001094.

[9] Gong, Boqing, Wei Lun Chao, Kristen Grauman e Fei Sha: *Diverse sequential subset selection for supervised video summarization*. Em Ghahramani, Z., M. Welling, C. Cortes, N. D. Lawrence e K. Q. Weinberger (editores): *Advances in Neural Information Processing Systems 27*, páginas 2069–2077. Curran Associates, Inc., 2014. http://papers.nips.cc/paper/5413-diverse-sequential-subset-selection-for-supervised-video-summarization.pdf.

[10] Mundur, Padmavathi, Yong Rao e Yelena Yesha: *Keyframe-based video summarization using delaunay clustering*. International Journal on Digital Libraries, 6(2):219–232, mar 2006. https://doi.org/10.1007/s00799-005-0129-9.

[11] Liu, D, Gang Hua e Tsuhan Chen: *A hierarchical visual model for video object summarization*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(12):2178–2190, dec 2010. https://doi.org/10.1109/tpami.2010.31.

[12] Lee, Yong Jae, J. Ghosh e K. Grauman: *Discovering important people and objects for egocentric video summarization*. Em *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2012. https://doi.org/10.1109/cvpr.2012.6247820.

[13] Ngo, Chong Wah, Yu Fei Ma e Hong Jiang Zhang: *Automatic video summarization by graph modeling*. Em *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 2003. https://doi.org/10.1109/iccv.2003.1238320.

[14] Laganière, Robert, Raphael Bacco, Arnaud Hocevar, Patrick Lambert, Grégory Païs e Bogdan E. Ionescu: *Video summarization from spatio-temporal features*. Em *Proceeding of the 2nd ACM workshop on Video summarization - TVS 08*. ACM Press, 2008. https://doi.org/10.1145/1463563.1463590.

[15] Nam, Jeho e Ahmed H. Tewfik. *Multimedia Tools and Applications*, 16(1/2):55–77, 2002. https://doi.org/10.1023/a:1013241718521.

[16] Hong, Richang, Jinhui Tang, Hung Khoon Tan, Shuicheng Yan, Chongwah Ngo e Tat Seng Chua: *Event driven summarization for web videos*. Em *Proceedings of the first SIGMM workshop on Social media - WSM 09*. ACM Press, 2009. https://doi.org/10.1145/1631144.1631154.

[17] Khosla, Aditya, Raffay Hamid, Chih Jen Lin e Neel Sundaresan: *Large-scale video summarization using web-image priors*. Em *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[18] Liu, Tiecheng e John R. Kender: *Optimization algorithms for the selection of key frame sequences of variable length*. Em *Computer Vision — ECCV 2002*, páginas 403–417. Springer Berlin Heidelberg, 2002. https://doi.org/10.1007/3-540-47979-1_27.

[19] Kang, Hong Wen, Xue Quan Chen, Y. Matsushita e Xiaoou Tang: *Space-time video montage*. Em *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR06)*. IEEE. https://doi.org/10.1109/cvpr.2006.284.

[20] Ma, Yu Fei, Lie Lu, Hong Jiang Zhang e Mingjing Li: *A user attention model for video summarization*. Em *Proceedings of the tenth ACM international conference on Multimedia - MULTIMEDIA 02*. ACM Press, 2002. https://doi.org/10.1145/641007.641116.

[21] Gygli, Michael, Helmut Grabner e Luc Van Gool: *Video summarization by learning submodular mixtures of objectives*. Em *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[22] Zhang, Ke, Wei Lun Chao, Fei Sha e Kristen Grauman: *Summary transfer: Exemplar-based subset selection for video summarization*. Em *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] Gygli, Michael, Helmut Grabner, Hayko Riemenschneider e Luc Van Gool: *Creating summaries from user videos.* Em *Computer Vision – ECCV 2014*, páginas 505–520. Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-10584-0_33.

[24] Chao, Wei Lun, Boqing Gong, Kristen Grauman e Fei Sha: *Large-margin determinantal point processes.* Em *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, páginas 191–200, Arlington, Virginia, United States, 2015. AUAI Press, ISBN 978-0-9966431-0-8. http://dl.acm.org/citation.cfm?id=3020847.3020868.

[25] *O Senhor dos Anéis: O Retorno do Rei.* https://www.imdb.com/title/tt0120737/?ref_=nmbio_trv_6. Último acesso: 2019-01-15.

[26] *Github dppLSTM.* https://github.com/kezhang-cs/Video-Summarization-with-LSTM. Último acesso: 2019-01-15.

[27] Kodratoff, Yves e Ryszard S. Michalski: *Machine Learning: An Artificial Intelligence Approach, Volume III.* Morgan Kaufmann, 1990, ISBN 9781558601192. https://www.amazon.com/Machine-Learning-Artificial-Intelligence-Approach/dp/1558601198?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1558601198.

[28] Mitchell, Tom M.: *Machine Learning.* McGraw-Hill Education, 1997, ISBN 0070428077. https://www.amazon.com/Machine-Learning-Tom-M-Mitchell/dp/0070428077?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0070428077.

[29] *Neuroscience.* Sinauer Associates Inc, 2004, ISBN 9780878937257. https://www.amazon.com/Neuroscience-Dale-Purves/dp/0878937250?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0878937250.

[30] Glorot, Xavier e Yoshua Bengio: *Understanding the difficulty of training deep feedforward neural networks.* Em *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.

[31] Rosenblatt, F.: *The perceptron: A probabilistic model for information storage and organization in the brain.* Psychological Review, 65(6):386–408, 1958. https://doi.org/10.1037/h0042519.

[32] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke e Andrew Rabinovich: *Going deeper with convolutions.* Em *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015. https://doi.org/10.1109/cvpr.2015.7298594.

[33] Hochreiter, Sepp e Jürgen Schmidhuber: *Long short-term memory*. Neural Computation, 9(8):1735–1780, nov 1997. https://doi.org/10.1162/neco.1997.9.8.1735.

[34] LeCun, Yann, Yoshua Bengio e Geoffrey Hinton: *Deep learning*. Nature, 521(7553):436–444, may 2015. https://doi.org/10.1038/nature14539.

[35] Howard, Jeremy e Sebastian Ruder: *Fine-tuned language models for text classification*. arXiv preprint arXiv:1801.06146, 2018.

[36] Lee, D. T. e B. J. Schachter: *Two algorithms for constructing a delaunay triangulation*. International Journal of Computer & Information Sciences, 9(3):219–242, jun 1980. https://doi.org/10.1007/bf00977785.

[37] Graves, A. e J. Schmidhuber: *Framewise phoneme classification with bidirectional LSTM networks*. Em *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* IEEE. https://doi.org/10.1109/ijcnn.2005.1556215.

[38] Bottou, Léon: *Large-scale machine learning with stochastic gradient descent*. Em *Proceedings of COMPSTAT2010*, páginas 177–186. Physica-Verlag HD, 2010. https://doi.org/10.1007/978-3-7908-2604-3_16.

[39] Buchbinder, Niv, Moran Feldman, Joseph Seffi e Roy Schwartz: *A tight linear time (1/2)-approximation for unconstrained submodular maximization*. SIAM Journal on Computing, 44(5):1384–1402, jan 2015. https://doi.org/10.1137/130929205.

[40] Potapov, Danila, Matthijs Douze, Zaid Harchaoui e Cordelia Schmid: *Category-specific video summarization*. Em *Computer Vision – ECCV 2014*, páginas 540–555. Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-10599-4_35.

[41] Song, Yale, Jordi Vallmitjana, Amanda Stent e Alejandro Jaimes: *Tvsum: Summarizing web videos using titles*. Em *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[42] *Dicionario Dicio*. https://www.dicio.com.br/estilo/. Último acesso: 2019-01-16.

[43] De Avila, Sandra Eliza Fontes, Ana Paula Brandão Lopes, Antonio da Luz Jr e Arnaldo de Albuquerque Araújo: *Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method*. Pattern Recognition Letters, 32(1):56–68, 2011.

[44] *Open video project*. http://www.open-video.org/. Último acesso: 2019-02-19.

[45] *Formato pickle*. https://docs.python.org/3/library/pickle.html.

[46] *Ubuntu*. https://www.ubuntu.com/. Último acesso: 2019-01-28.

[47] *Jupyter*. https://jupyter.org/. Último acesso: 2019-01-28.

[48] *Python 2.7*. https://docs.python.org/2/index.html. Último acesso: 2019-01-28.

[49] *Numpy*. https://www.numpy.com/. Último acesso: 2019-01-28.

[50] *H5py*. https://www.h5py.com/. Último acesso: 2019-01-28.

[51] *Scipy*. https://www.scipy.org/. Último acesso: 2019-01-28.

[52] *Matplotlib*. https://matplotlib.org/. Último acesso: 2019-01-28.

[53] *Base jumping com wingsuit*. https://www.youtube.com/watch?v=Mun_kUi9dx4.

[54] *Urban Mountain Biking*. https://www.youtube.com/watch?v=mFuSjk7jv_M.