



UNIVERSIDADE DO ESTADO DO
RIO DE JANEIRO

INSTITUTO POLITÉCNICO
GRADUAÇÃO EM ENGENHARIA
DE COMPUTAÇÃO



Thiago Cler Franco

Detecção de créditos finais em produtos audiovisuais utilizando
inteligência computacional

Nova Friburgo
2021



UNIVERSIDADE DO ESTADO DO
RIO DE JANEIRO

INSTITUTO POLITÉCNICO
GRADUAÇÃO EM ENGENHARIA
DE COMPUTAÇÃO



Thiago Cler Franco

**Detecção de créditos finais em produtos audiovisuais utilizando inteligência
computacional**

Trabalho de conclusão de curso apresentado
como pré-requisito para obtenção do título
de Engenheiro de Computação, ao Departamento de Modelagem Computacional, do
Instituto Politécnico, da Universidade do Es-
tado do Rio de Janeiro.

Orientador: Prof. Dr. Bernardo Sotto-Maior Peralva

Nova Friburgo

2021

UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
INSTITUTO POLITÉCNICO - CURSO DE ENGENHARIA DE COMPUTAÇÃO

Reitor: Ricardo Lodi Ribeiro

Vice-Reitor: Mario Sérgio Alves Carneiro

Diretor do Instituto Politécnico: Angelo Mondaini Calvão

Coordenadora de Curso: Ana Cristina Fontes Moreira

Banca Avaliadora Composta por: Prof. Dr. Bernardo Sotto-Maior Peralva (Orientador)

Prof. Dr. Francisco Duarte Moura Neto

Prof. Dr. Roberto Pinheiro Domingos

Ficha elaborada pelo autor através do
Sistema para Geração Automática de Ficha Catalográfica da Rede Sirius - UERJ

F825 Franco, Thiago Cler
 Detecção de créditos finais em produtos
 audiovisuais utilizando inteligência computacional /
 Thiago Cler Franco. - 2021.
 58 f.

Orientador: Bernardo Sotto-Maior Peralva
Trabalho de Conclusão de Curso apresentado à
Universidade do Estado do Rio de Janeiro, Instituto
Politécnico, para obtenção do grau de bacharel em
Engenharia da Computação.

1. Créditos de Encerramento - Monografias. 2.
Extração de Metadados de Vídeos - Monografias. 3.
Streaming - Monografias. 4. Aprendizado de Máquina -
Monografias. I. Peralva, Bernardo Sotto-Maior. II.
Universidade do Estado do Rio de Janeiro. Instituto
Politécnico. III. Título.

CDU 004.41

Endereço: UERJ - IPRJ, Rua Bonfim, 25 - Prédio 5, Vila Amélia. CEP 28625-570
- Nova Friburgo - RJ - Brasil.

Este trabalho nos termos da legislação que resguarda os direitos autorais é considerado de propriedade da Universidade do Estado do Rio de Janeiro (UERJ). É permitida a transcrição parcial de partes do trabalho, ou mencioná-lo, para comentários e citações, desde que sem propósitos comerciais e que seja feita a referência bibliográfica completa.

Thiago Cler Franco

Detecção de créditos finais em produtos audiovisuais utilizando inteligência computacional

Trabalho de conclusão de curso apresentado como pré-requisito para obtenção do título de Engenheiro de Computação, ao Departamento de Modelagem Computacional, do Instituto Politécnico, da Universidade do Estado do Rio de Janeiro.

Aprovado em 06 de Agosto de 2021.

Banca Examinadora:

Prof. Dr. Bernardo Sotto-Maior Peralva (Orientador)
Instituto Politécnico – UERJ

Prof. Dr. Francisco Duarte Moura Neto
Instituto Politécnico – UERJ

Prof. Dr. Roberto Pinheiro Domingos
Instituto Politécnico – UERJ

Nova Friburgo
2021

DEDICATÓRIA

Este trabalho é dedicado a meus pais,
que tanto se dedicaram para que a trajetória que culmina nestas páginas fosse possível.

AGRADECIMENTOS

À Romária e Sebastião, meus pais, por todo amor e educação que me proporcionaram.

À minha família, pelo amor e apoio.

Aos meus amigos, pelo suporte e incentivo.

À UERJ, por me proporcionar oportunidades de aprendizado, oportunidades profissionais e, sobretudo, oportunidade de conhecer pessoas incríveis.

Ao meu orientador, Bernardo, por ter aceitado me guiar neste trabalho.

RESUMO

Franco, Thiago Cler. *Detecção de créditos finais em produtos audiovisuais utilizando inteligência computacional.* 2021. 58 f. Trabalho de Conclusão de Curso (Engenharia de Computação) – Instituto Politécnico do Rio de Janeiro, Universidade do Estado do Rio de Janeiro, Nova Friburgo, 2021.

Plataformas de *streaming* de vídeos fornecem funcionalidades de interação para enriquecer a experiência do usuário ao assistir um conteúdo. Uma dessas funcionalidades é a de “pular créditos”, que possibilita ao usuário interromper a reprodução de um vídeo quando o mesmo atinge o momento de exibição dos créditos finais da obra e, imediatamente, iniciar a reprodução de um novo conteúdo, em geral relacionado ao anterior. Tal funcionalidade permite que o fluxo de consumo na plataforma seja facilitado, gerando melhor experiência para os usuários e, consequentemente, melhor retorno para o provedor. Para que essa funcionalidade seja possível, porém, é necessário que sejam identificados e cadastrados os tempos de início dos créditos dos vídeos da plataforma. Se feito de forma manual, esse processo pode ser muito custoso. Este trabalho visa elaborar métodos para automatizar a identificação do momento de início dos créditos em vídeos de séries e filmes. Para isso, a partir de dados de uma plataforma de *streaming* real, foram desenvolvidos modelos baseados em visão computacional e aprendizado de máquina para realizar a tarefa em questão. Os modelos foram avaliados comparando-se seu resultado com os momentos de início de créditos marcados manualmente.

Palavras-chave: Extração de Metadados de Vídeos. Aprendizado de Máquina. Visão Computacional. Créditos de Encerramento em Vídeos.

ABSTRACT

Franco, Thiago Cler. *Detection of ending credits on audiovisual products using computational intelligence*. 2021. 58 f. Trabalho de Conclusão de Curso (Engenharia de Computação) – Instituto Politécnico do Rio de Janeiro, Universidade do Estado do Rio de Janeiro, Nova Friburgo, 2021.

Video streaming platforms provide interaction functionalities to improve user experience when watching a content. One of these functionalities is the “skip credits” one, that allows the user to interrupt the playback of a video when it makes to the moment of exhibition of ending credits and immediatly start a new content, generally related to the previous one. Such funtionality allows the platform’s consumption flow to be made easier, creating a better user experience and better results to the service provider. For this feature to work, it is necessary to register the start time of the ending credits for the platform’s videos, which can be irksome to be done manually. This study aims to elaborate methods that automate the identification of start time of credits on series and movies. This is done by using real data from a streaming platform to develop models based on computer vision and machine learning capable of such identification. The models are evaluated through comparision with manually registered start times of credits on videos.

Keywords: Video Metadata Extraction. Machine Learning. Computer Vision. Videos
End Credits.

LISTA DE FIGURAS

Figura 1 - Representação Matricial dos Quadros de um Vídeo	17
Figura 2 - Arquitetura de uma Plataforma de <i>Streaming</i>	18
Figura 3 - Interfaces de plataformas de <i>streaming</i>	19
Figura 4 - Interfaces de <i>players</i> de vídeo	20
Figura 5 - Funcionalidade pular créditos	21
Figura 6 - Exemplos de <i>frames</i> pertencentes à diferentes sequências de créditos . .	23
Figura 7 - Exemplos de sequências de créditos não suportadas	24
Figura 8 - Jupyter Notebook	26
Figura 9 - Transição de fim do conteúdo para o início dos créditos	29
Figura 10 - Subtração de imagens	30
Figura 11 - Reorganização de matrizes em um vetor	30
Figura 12 - Derivada da normas das diferenças entre <i>frames</i>	32
Figura 13 - Identificação gráfica do momento de início dos créditos	33
Figura 14 - Segmentação de regiões da imagem que contêm letras	37
Figura 15 - Exemplo de árvore de decisão	40
Figura 16 - Exemplo de rede neural multicamadas	41
Figura 17 - Importância das <i>features</i> para o modelo de floresta aleatória	44
Figura 18 - Matriz de confusão da floresta aleatória com dados desbalanceados . .	46
Figura 19 - Matriz de confusão da floresta aleatória com dados balanceados . . .	47
Figura 20 - Matriz de confusão da rede neural com dados desbalanceados . . .	47
Figura 21 - Matriz de confusão da floresta aleatória com dados balanceados . . .	48
Figura 22 - Análise gráfica da detecção usando floresta aleatória com dados desbalanceados	50
Figura 23 - Análise gráfica da detecção usando floresta aleatória com dados balanceados	52
Figura 24 - Análise gráfica da detecção usando redes neural com dados desbalanceados	53
Figura 25 - Análise gráfica da detecção usando rede neural com dados balanceados	54

LISTA DE TABELAS

Tabela 1 - Estatísticas descritivas do erro da técnica de diferença entre <i>frames</i> . . .	43
Tabela 2 - Total de marcações aceitáveis	43
Tabela 3 - Métricas de acurácia e <i>F1-score</i> dos modelos	45
Tabela 4 - Métricas de desempenho da floresta aleatória com dados desbalanceados	46
Tabela 5 - Métricas de desempenho da floresta aleatória com dados balanceados .	47
Tabela 6 - Métricas de desempenho da rede neural com dados desbalanceados . .	48
Tabela 7 - Métricas de desempenho da rede neural com dados balanceados	49
Tabela 8 - Estatísticas descritivas do erro nos modelos de classificadores de <i>frames</i>	50

LISTA DE ABREVIATURAS E SIGLAS

ABR	<i>Adaptative Bitrate Streaming</i>
CDN	<i>Content Delivery Network</i>
FPS	<i>Frames per Second</i>
HTTP	<i>Uniform Resource Locator</i>
IPRJ	Instituto Politécnico
OTT	<i>Over The Top</i>
RGB	<i>Red Green Blue</i>
UERJ	Universidade do Estado do Rio de Janeiro
UX	<i>User Experience</i>
VoD	<i>Video on Demand</i>

SUMÁRIO

	INTRODUÇÃO	13
1	PLATAFORMAS DE <i>STREAMING</i> DE VÍDEOS	16
1.1	Vídeo Digital	16
1.2	<i>Streaming</i>	17
1.3	Metadados de Vídeos	18
1.4	<i>Cuepoints</i>	19
1.5	<i>Player</i> de Vídeo	19
2	DETECÇÃO MANUAL DO ÍNICO DE CRÉDITOS	22
2.1	Cadastro manual de metadados	22
2.2	Créditos de encerramento	22
2.3	Escopo	23
3	DETECÇÃO AUTOMÁTICA DO INÍCIO DE CRÉDITOS	25
3.1	Ambiente de desenvolvimento	26
3.2	Obtenção de dados	27
3.3	Modelo de detecção por derivada da diferença entre frames	28
3.3.1	Diferença entre <i>frames</i> consecutivos	29
3.3.2	Derivada da diferença entre <i>frames</i> consecutivos	31
3.3.3	Obtenção do tempo de início dos créditos	32
3.4	Modelo de detecção por classificador de frames	33
3.4.1	Inteligência Artificial	34
3.4.2	Aprendizado de máquina	34
3.4.3	Solução proposta	35
3.4.4	Dados de treino	35
3.4.5	Seleção dos parâmetros de entrada	36
3.4.5.1	Ordem do frame	36
3.4.5.2	Norma da diferença entre <i>frames</i>	36
3.4.5.3	Norma da diferença entre histogramas	37
3.4.5.4	Número de letras	37
3.4.5.5	Fluxo óptico	38
3.4.6	Pré-processamento dos dados	38
3.4.7	Classificador	39
3.4.7.1	Floresta Aleatória	39
3.4.7.2	Redes Neurais	40
4	RESULTADOS	42
4.1	Diferença entre <i>frames</i>	42
4.2	Classificação de <i>frames</i>	44

4.3	Detecção de créditos usando classificador	50
4.4	Pesquisa Reproduzível	54
	CONCLUSÃO	55
	REFERÊNCIAS	57

INTRODUÇÃO

Registros por meio de imagens cumprem um papel fundamental no desenvolvimento da espécie humana. Desde pinturas rupestres até as mais recentes obras cinematográficas, a representação visual do homem e sua relação com o meio tem servido a diferentes propósitos. De imediato, comunica - transmite uma mensagem contendo o retrato de um determinado povo, lugar e sua interação. Quando essa comunicação transcende o tempo, registros históricos são criados, alimentando o imaginário coletivo. Em uma instância abstrata, trata-se de arte. Quando associada ao objetivo de divertir ou distrair, torna-se entretenimento.

A ação de se observar imagens em movimento sendo exibidas em uma tela está presente, hoje, em grande parte das atividades desempenhadas pelas pessoas, em particular quando buscam se entreter. A partir da televisão e do cinema, o consumo de vídeos tornou-se cada vez mais popular. Foi incorporado ao dia a dia até tornar-se um dos principais recursos de lazer existentes, com a expectativa de que essa atividade seja responsável por 82% do tráfego na internet até 2022 (CISCO, 2018). A alta demanda proporcionou o surgimento de grandes mercados de entretenimento, os quais se fizeram valer do advento de novas tecnologias para manterem-se em constante evolução, gerando conteúdos cada vez mais acessíveis e plurais. O consumo de vídeo partiu de uma estratégia centralizada de distribuição na qual o consumidor final tinha pouca influência sobre o que assistiria, se abrigou em diferentes mídias ao longo do tempo - da fita cassete ao vhs, do DVD ao blu-ray - até que encontrou a internet. Nesse meio, o controle passa às mãos de quem assiste. Essa mudança de paradigma é um catalisador para uma transformação na forma de se interagir com séries, filmes, programas de variedades ou quaisquer outras das inúmeras classes de conteúdo possíveis.

A internet foi palco para o surgimento das chamadas OTTs (*Over the Top*), empresas que oferecem uma plataforma para consumo de vídeos sob demanda (como Netflix, HBO Max, Disney+, Globoplay) com modelo de negócios D2C (*direct to consumer*) que dispensa o intermédio de empresas de telecomunicações, tais quais operadoras de TV a cabo, oferecendo seus serviços diretamente ao consumidor. O espectador não mais precisa se submeter a grades de programação com horários e conteúdos engessados - assiste ao que quiser, onde e quando desejar.

Essa liberdade fomentou a adoção de inúmeras novas formas de consumo, totalmente delineadas pelas preferências e hábitos dos usuários. Enquanto um pode optar por assistir todos os episódios de uma série em sequência durante um único dia, outro pode preferir assistir a um episódio por semana. Uma dessas formas de consumo, entretanto, destaca-se por representar não apenas comodidade aos usuários, mas também por trazer benefícios às plataformas que proveem o serviço de distribuição de vídeos *online*. Trata-se

do chamado *binge-watching*, em que o conteúdo é assistido em formato de "maratona", com um episódio logo após o outro (STEINER, 2018). O *binge-watching* potencializa o engajamento dos usuários dentro e fora da plataforma, traduzindo-se em fidelização e publicidade. Conforme (AMERI; HONKA; XIE, 2019), esse comportamento intenso torna mais provável, inclusive, o consumo de conteúdos derivados ou sequências.

As plataformas OTT, em busca de vantagens competitivas e maior engajamento dos usuários, empalam-se em oferecer funcionalidades que tornem a experiência de consumo de vídeo cada vez melhor. Dados os benefícios do modelo de *binge-watching*, é natural que existam funcionalidades dedicadas à sua facilitação. Uma dessas funcionalidades, em específico, se aproveita do fato de que diversos programas, filmes e séries costumam ter uma característica em comum: nos minutos finais do vídeo, são exibidas telas que apresentam informações textuais creditando os profissionais envolvidos na obra (os créditos finais ou créditos de encerramento). O espectador pode não ter interesse em assistir a sequência de créditos e preferir parar de acompanhar o vídeo ou buscar outro conteúdo para seguir assistindo imediatamente. Nesse contexto, uma funcionalidade que vem sendo explorada é a de "pular créditos", que consiste em dar ao usuário, no momento em que se inicia a sequência de créditos do programa, a opção de começar a assistir a partir daquele momento um próximo conteúdo.

Objetivo

Para que a funcionalidade de "pular créditos" seja possível, é necessário que seja previamente conhecido o momento do vídeo no qual a sequência de créditos se inicia. Essa é uma informação, a priori, custosa de ser obtida, pois demanda esforço manual - um ser humano deve, para cada filme ou episódio de programa, identificar esse momento de interesse no vídeo e registrar essa informação.

Com vistas à redução do esforço empenhado para a disponibilização da referida funcionalidade em plataformas OTT, este trabalho tem como objetivo o estudo e aplicação de técnicas computacionais que possibilitem a identificação do momento de início da sequência de créditos em vídeos de forma automática, eliminando a dependência de processos manuais. Sua contribuição é propor a implementação de uma técnica eficaz e que possa ser aplicada tanto na resolução do problema em questão como em quaisquer outras situações que se beneficiem da informação do momento de início de créditos em filmes e séries.

Estrutura do Trabalho

O Capítulo 1 destina-se à introdução de conceitos básicos necessários à compreensão deste texto. São apresentadas informações relacionadas ao contexto de consumo de vídeos sob demanda em plataformas *online*.

No Capítulo 2 é descrito o problema da identificação do momento de início da sequência de créditos finais em vídeos. São detalhados os processos envolvidos na identificação manual desses pontos de interesse, as características das sequências de créditos a serem tratadas e a proposta para a identificação automática.

O Capítulo 3 traz o desenvolvimento das soluções propostas, desde o fundamento teórico até a implementação.

O Capítulo 4 é responsável pela apresentação, análise e comparação dos resultados obtidos com as soluções desenvolvidas.

A conclusão do trabalho é feita no Capítulo 5, avaliando seus resultados, contribuições e possíveis oportunidades de evolução.

1 PLATAFORMAS DE *STREAMING* DE VÍDEOS

Aqui são apresentados os principais conceitos técnicos envolvidos no ecossistema de OTTs, com foco nos elementos mais importantes dentro do fluxo de consumo de vídeos pelo usuário. Partindo da definição de vídeo digital, o texto evolui abordando processo de transmissão até tratar dos mecanismos de interação do usuário com a mídia.

1.1 Vídeo Digital

Ao evoluir de uma representação analógica para uma representação digital, sinais de vídeo ganharam inúmeras possibilidades de utilização. Esses sinais são não só representados, como também armazenados e transmitidos digitalmente (DHANANI; PARKER, 2012). Esta representação digital se baseia em uma sequência de imagens, ou *frames*, que contém um determinado número de pontos, ou *pixels*.

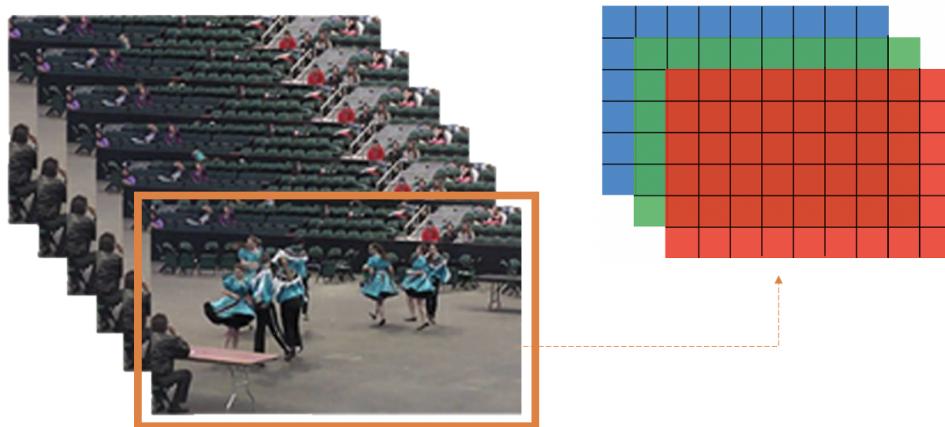
Vídeos digitais podem se apresentar com diferentes números de *frames* por segundo (fps). Enquanto filmes, por exemplo, costumam ser gravados em 24 fps, taxas de 30 ou 60 fps também são amplamente utilizadas, onde valores maiores podem levar a maior percepção de movimento, sendo ideais para conteúdos mais dinâmicos. Além disso, o número de *pixels* presente nos *frames* determina a resolução do vídeo. A resolução é descrita pela quantidade de *pixels* horizontais versus a quantidade de *pixels* verticais. Maiores resoluções levam a maiores níveis de detalhes nas imagens. Exemplos típicos de resolução são 360p (640 x 360), 720p (1260 x 720) e 1080p (1920 x 1080).

Cada *pixel* de um *frame*, por sua vez, assume uma determinada cor que é dada pela combinação das cores vermelho, verde e azul - que originam o sistema de cores RGB (do inglês: *red*, *green*, *blue*). Um *pixel* pode ser representado por um vetor de três componentes, cada uma indicando a contribuição das respectivas cores primárias do RGB. Em termos computacionais, a quantidade de cores que um *pixel* é capaz de assumir dependerá da quantidade de memória alocada para a representação do pixel. Uma representação que utilize 8 bits para representar cada componente de cor será capaz de gerar $2^8 = 256$ valores possíveis de vermelho, verde e azul respectivamente. Apesar do sistema RGB ser amplamente utilizado em vídeos digitais, outros espaços de cores também podem ser citados, como o YCrCb que codifica o valor referente à luminância do pixel, além das informações de cores.

Para representar todas as características de um vídeo digital como descritas até aqui, é comumente utilizada uma abordagem matricial. Tomando um vídeo com 20 segundos de duração, de resolução 1920 x 1080 com 30 fps e no espaço de cores RGB, sua representação pode ser dada por uma matriz de 4 dimensões: a primeira referente ao

número de colunas de *pixels* presente em cada *frame* e a segunda referente ao número de linhas, a terceira em relação às componentes de cores e a última denotando o número de *frames* do vídeo. No exemplo em questão teria-se uma matriz 1920 x 1080 x 3 x 600. Essa representação é ilustrada na Figura 1.

Figura 1 - Representação Matricial dos Quadros de um Vídeo



Um vídeo, dado por uma sequência de quadros, cada um sendo representado por uma matriz de três dimensões. Fonte: O autor, 2021

1.2 *Streaming*

As plataformas OTT de interesse, também chamadas plataformas de *streaming*, possibilitam que seus clientes tenham acesso a diversos conteúdos em formato de vídeo digital através de dispositivos (como computadores, smartphones e tvs) conectados à internet. Para isso, precisam transmitir os vídeos armazenados em seus servidores pela rede e prover software capaz de reproduzir os vídeos recebidos no dispositivo do usuário. Tal processo é o que denota-se por *streaming*.

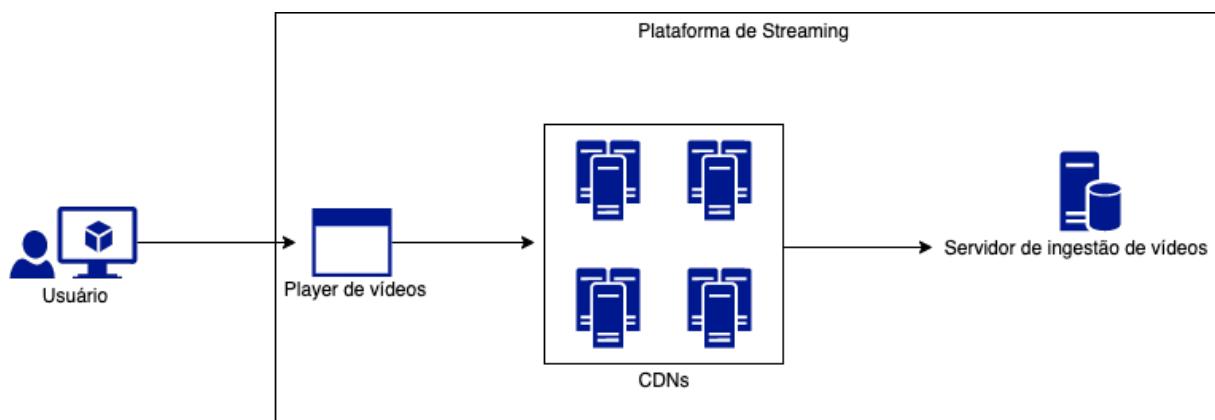
Em linhas gerais, o processo de *streaming* compreende algumas etapas, desde o processamento de vídeos, sua distribuição até sua reprodução (WU Y. T. HOU; PEHA, 2001). Considerando o cenário de *streaming* de vídeos sob demanda, ou VoD (*Video on Demand*), a primeira etapa corresponde ao *encoding* do vídeo original a ser transmitido. Tal etapa pode ser entendida como uma fase de compactação, cujo objetivo é reduzir o tamanho do arquivo de vídeo mantendo ao máximo sua qualidade, de modo a facilitar o tráfego de dados. Nesta etapa também são geradas todas as qualidades de vídeo que estarão disponíveis para o usuário.

Um segundo momento é quando um vídeo é solicitado aos servidores, o que ocorre, por exemplo, por meio de uma requisição que é enviada quando um usuário seleciona um

vídeo para assistir. Comumente, a plataforma de *streaming* possui uma rede de servidores chamada CDN (*Content Delivery Network*, ou rede de entrega de conteúdo) presente em regiões geográficas possivelmente distintas, responsável por responder às requisições com os arquivos necessários para a reprodução dos vídeos. As CDNs podem otimizar as entregas de acordo com fatores como a localização de origem da requisição e o tipo de conteúdo solicitado. A entrega do vídeo é feita usando a tecnologia ABR (*Adaptive Bitrate Streaming*), que consiste em mapear o vídeo a ser entregue em segmentos e entregar esses segmentos progressivamente. Isso permite que um vídeo, disponível nas qualidades 360p, 720p e 1080p, por exemplo, possa ser reproduzido em qualquer uma das resoluções a qualquer momento de forma intercambiável e mais adequada à largura de banda disponível, bastando à CDN entregar os segmentos correspondentes à qualidade solicitada.

Por fim, o componente do sistema de *streaming* com o qual o usuário interage diretamente é o *player*. Ele é o software responsável por solicitar à CDN os segmentos da mídia que precisa ser reproduzida, bem como interpretar os segmentos recebidos e sua codificação para então tocar o vídeo. A Figura 2 ilustra uma arquitetura encontrada em plataformas de *streaming*.

Figura 2 - Arquitetura de uma Plataforma de *Streaming*



Arquitetura típica de uma plataforma OTT. Usuário interage com o *player* de vídeos, que solicita à CDN os recursos de mídia a serem reproduzidos. Fonte: O autor 2021

1.3 Metadados de Vídeos

No *streaming* de vídeos, além dos dados necessários à reprodução das mídias, como os arquivos de vídeo em si, existem outros dados que também precisam ser transmitidos. Tais dados têm um caráter complementar, enriquecem o conteúdo presente no vídeo digital com informações que podem tornar a experiência do consumo mais atrativa. São os chamados metadados do vídeo. Dentre os diversos tipos de metadados existentes, podem

ser agrupados aqueles que têm um significado imediato para o usuário e aqueles cuja relevância se dá pelo apoio a funcionalidades do sistema de transmissão ou reprodução de mídia. No primeiro grupo, frequentemente figuram informações como título da obra, responsáveis (autores, produtores, diretores), elenco, ano de criação/exibição, gênero, classificação etária. Já o segundo grupo de metadados - no qual está o foco deste trabalho - engloba informações que, por si só, não necessariamente têm algum valor mas, quando aplicadas ao contexto do *streaming*, possibilitam uma gama de experiências; na subseção a seguir um importante metadado deste grupo é explorado.

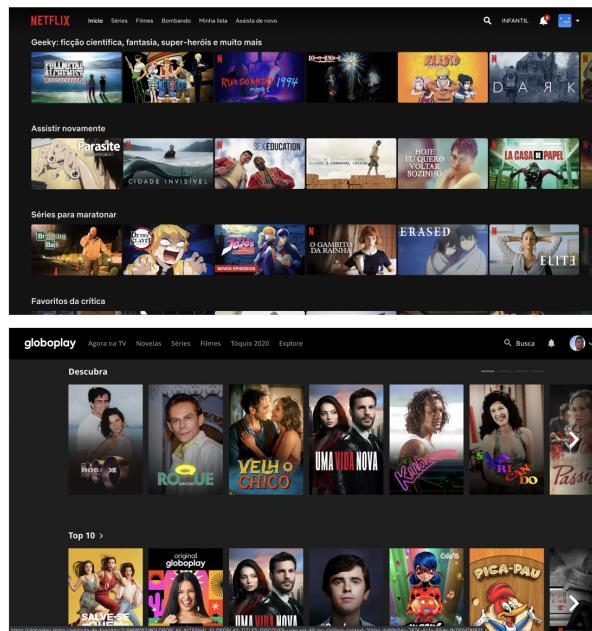
1.4 *Cuepoints*

Uma categoria importante de metadados de vídeos é a de pontos de interesse, os chamados *cuepoints*. Os pontos de interesse marcam momentos dentro da cronologia do vídeo que possuem uma relevância intrínseca. Eles correspondem ao tempo exato de um acontecimento interno ou externo ao conteúdo da mídia e são usados como gatilho para o disparo de determinados eventos. Podem ser exemplos de *cuepoints* instantes do vídeo em que ocorram trocas de cena ou de conteúdo ou momentos predefinidos pela plataforma para a exibição de anúncios. Outro exemplo é o tempo no vídeo onde começam e terminam de ser exibidas vinhetas e trechos comuns a mais de uma mídia, como costumeiramente é o caso de sequências de abertura em novelas e séries. De forma análoga, um último exemplo - e objeto de estudo deste trabalho - é o momento em que se inicia a sequência de créditos de encerramento em filmes e séries.

1.5 *Player* de Vídeo

Como citado anteriormente, o *player* é o ponto de contato entre o usuário e a mídia à qual deseja-se fazer o *streaming*. Costumeiramente, a interação inicial do usuário com a plataforma OTT se dá através de uma interface - ilustrada na Figura 3 - onde estão dispostos os conteúdos disponíveis para o consumo. Quando ele tomar uma ação que expresse a intenção de assistir um dos conteúdos (por exemplo, um clique em um cartaz de um filme), passará a ver na tela a interface do *player* de vídeos. Essa interface pode variar de acordo com as definições de UX (*user experience*, ou experiência de usuário) de cada provedor mas, em geral, possui um botão de *play/pause*, ajuste de volume, seletor de configurações como idioma, legenda ou qualidade e um *scrubber* - barra indicativa do progresso do vídeo na qual é possível avançar ou retroceder a reprodução. Exemplos de interfaces de *players* de vídeos são apresentados na Figura 4.

Figura 3 - Interfaces de plataformas de *streaming*



Interfaces das plataformas Netflix (acima) e globoplay (abaixo). Fonte: Captura de tela realizada pelo autor 2021

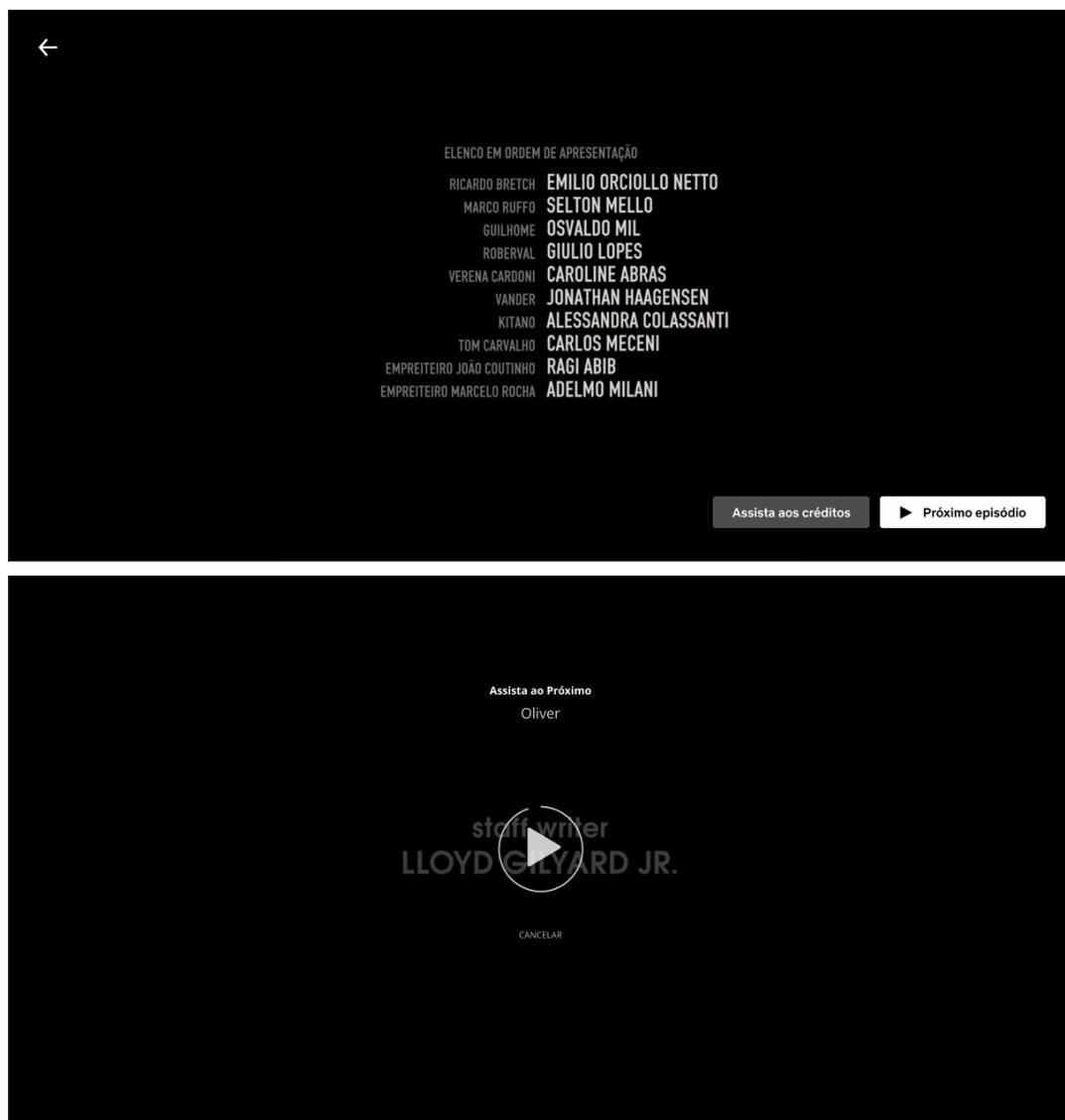
Figura 4 - Interfaces de *players* de vídeo



Player de vídeos da Netflix (acima) e do globoplay (abaixo). Fonte: Captura de tela realizada pelo autor, 2021

Além das funcionalidades básicas descritas, o *player* pode prover outras experiências de consumo e interação. Em especial, a partir dos metadados associados ao vídeo em reprodução, pode exibir informações e propor ações ao usuário. Nesse contexto está incluída a funcionalidade de “pular créditos”. O *player* tem acesso aos metadados de *cuepoints* do vídeo, dentre os quais pode existir o ponto de início dos créditos no conteúdo atual. Com essa informação, quando a reprodução da mídia alcançar o momento indicado pelo *cuepoint*, o *player* pode exibir um *feedback* para o usuário, usualmente na forma de um botão, sugerindo que ele comece a assistir um novo conteúdo imediatamente. Esse cenário é exemplificado na Figura 5.

Figura 5 - Funcionalidade pular créditos



Diferentes *players* apresentando função de “pular créditos”. São apresentadas opções de assistir ao próximo episódio ou continuar assistindo aos créditos. Fonte: Captura de tela realizada pelo autor, 2021

2 DETECÇÃO MANUAL DO ÍNICO DE CRÉDITOS

Neste capítulo é detalhado o problema ao qual destina-se a solução proposta neste trabalho. Considera-se aqui uma plataforma de *streaming* que possua as características descritas anteriormente. Nesse cenário, um dos fluxos operacionais existentes é o de ingestão dos vídeos que serão disponibilizados na plataforma. Este fluxo engloba desde o *upload* de arquivos de mídia (como o próprio vídeo, faixas de áudio e legendas) até o cadastro de metadados. Partes desse fluxo podem ou não ser automatizadas, de acordo com sua relevância, urgência dentro do processo, tolerância à falhas, níveis de eficácia e eficiência da automação e a capacidade da plataforma de fazê-lo.

2.1 Cadastro manual de metadados

Metadados podem tratar de inúmeros tipos de informação, muitos dos quais oriundos de diferentes fontes. Ademais, podem ser usados de forma editorial, atendendo a expectativas e estratégias de negócio. São essenciais para a plataforma de *streaming*.

Dada a importância dos metadados, é comum que existam pessoas dedicadas ao seu fluxo de cadastro. Essas pessoas, chamadas editoras, possuem acesso às informações que devem ser cadastradas (quando tais informações são provenientes de fontes externas) ou então realizam elas próprias os processos necessários para a obtenção das mesmas. Um desses processos pode ser efetivamente assistir ao vídeo de origem. As editoras costumam ter à sua disposição aparato que facilite a reprodução de vídeos, como salas com múltiplos monitores além de *mouses* ou outras interfaces facilitadas de interação.

Determinados metadados requerem a atenção do editor para serem obtidos, como é o caso de alguns cuepoints. Um editor pode precisar assistir repetidas vezes um mesmo conteúdo a fim de encontrar pontos de interesse específicos. É o que acontece com os pontos de início dos créditos de encerramento. A pessoa editora precisa se atentar ao momento no qual os créditos começam a ser exibidos e, então, resgatar o tempo correspondente no vídeo e fazer o cadastro dessa informação.

2.2 Créditos de encerramento

Aqui define-se como créditos de encerramento uma sequência contínua de *frames* que se passa durante os momentos finais da duração de um vídeo, apresentando informações relativas à produção da obra. Essas informações frequentemente estão dispostas no formato de texto, com descrições de funções ou cargos próximos aos nomes dos

profissionais creditados. Também frequentemente, esses blocos de texto são dispostos sob um fundo preto estático. Algumas sequências de créditos podem apresentar uma composição mais complexa, com animações e conteúdo mais dinâmico. Exemplos de *frames* pertencentes a sequências de créditos são apresentados na Figura 6. Encontrar o início dos créditos pode se tornar uma tarefa extremamente repetitiva, considerando-se que o fluxo de cadastro será realizado para cada episódio de cada temporada de cada programa, além de filmes que forem adicionados à plataforma. Como tal, é um processo passível de ser automatizado. Fazer com que o sistema seja capaz de identificar o *cuepoint* de início dos créditos poupa ao editor um tempo que poderá ser empregado em outras tarefas, otimizando o fluxo.

Figura 6 - Exemplos de *frames* pertencentes à diferentes sequências de créditos



Fonte: Captura de tela realizada pelo autor, 2021

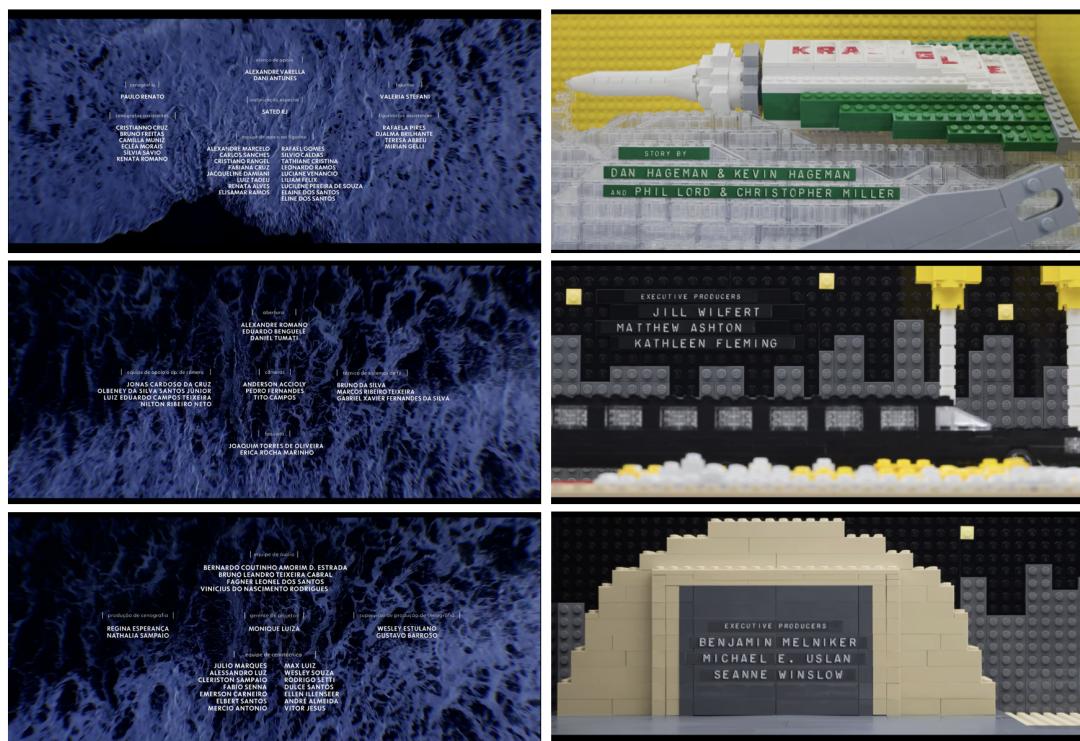
2.3 Escopo

O problema tratado neste trabalho é a identificação manual do momento de início dos créditos em vídeos. A solução proposta consiste na identificação automática do momento de início dos créditos. Antes do desenvolvimento da solução, é necessário apresentar

o escopo do projeto. Um escopo precisa ser bem definido para que os resultados possam ser obtidos e mensurados de maneira assertiva. Dessa forma, são estabelecidas as expectativas e limitações do projeto:

- A solução deve ser capaz de identificar sequências de crédito que possuam algumas características predefinidas;
 - A solução deve receber como entrada o caminho para um arquivo de vídeo e retornar como saída o tempo em milissegundos correspondente ao momento de início dos créditos no vídeo;
 - As sequências de créditos a serem identificadas devem possuir fundo estático, preferencialmente preto;
 - As sequências de créditos a serem identificadas devem apresentar apenas conteúdo textual;
 - Não são atendidas pela solução sequências de créditos que apresentem conteúdos dinâmicos, visualmente complexos ou que não atendam às limitações anteriores.
- Exemplos de sequências não suportadas na Figura 7

Figura 7 - Exemplos de sequências de créditos não suportadas



À esquerda, sequência que apresenta conteúdo textual comum, porém com fundo dinâmico, uma cena do mar em movimento; à direita, sequência composta por grande variação de elementos visuais complexos. Fonte: Captura de tela realizada pelo autor, 2021

3 DETECÇÃO AUTOMÁTICA DO INÍCIO DE CRÉDITOS

Definido o problema, nesta seção é apresentada a solução proposta. São descritos os fundamentos necessários à construção da solução, em âmbito teórico e prático. Partindo de uma breve recapitulação do problema, são estabelecidos os paralelos entre as atividades de caráter manual envolvidas no processo em questão e as possíveis automações correspondentes. A metodologia seguida no desenvolvimento é explanada e, então, são detalhadamente dispostas todas as etapas para a obtenção dos resultados almejados.

Tem-se por objetivo o desenvolvimento de uma solução capaz de, dado um vídeo que contém um crédito nos padrões anteriormente estabelecidos, identificar o momento de início da sequência de créditos. Em um processo manual, a pessoa editora precisa inicialmente reproduzir o vídeo em questão. Em seguida, aproveitando-se da característica intrínseca aos créditos de encerramento, ela poderá avançar a reprodução do vídeo para seus últimos minutos. A partir de então, podendo ou não estar apoiada por funcionalidade de pré-visualização de *frames* ao navegar pelo *scrubber* do *player*, a editora buscará identificar onde se iniciam os créditos, avançando ou retrocedendo no vídeo o quanto for necessário. Neste momento, sua atenção estará voltada a encontrar *frames* que contenham, ao invés de cenas, informações textuais, bem como momentos em que ocorram transições abruptas para *frames* pretos e estáticos. Essa busca pode ser apoiada também pelo áudio da mídia, o qual geralmente destoa do restante do conteúdo pela ausência de diálogos e eventual trilha sonora marcante. Após o primeiro *frame* com características adequadas ser observado, o editor analisará *frames* anteriores com o intuito de precisar o exato *frame* no qual a sequência se inicia. Por fim, será necessário que se observe o tempo do vídeo no qual o *frame* ocorre e posteriormente fazer o registro desse tempo.

A proposta de automação pretende simplificar o processo ao ponto que todo o fluxo, desde a reprodução do vídeo até a observação do tempo de início dos créditos, torne-se dispensável. Em substituição, o vídeo é usado como entrada para um algoritmo que o processará e, com base em algumas das mesmas características observadas pelos editores, identificará e retornará o tempo que indica o início da sequência de créditos.

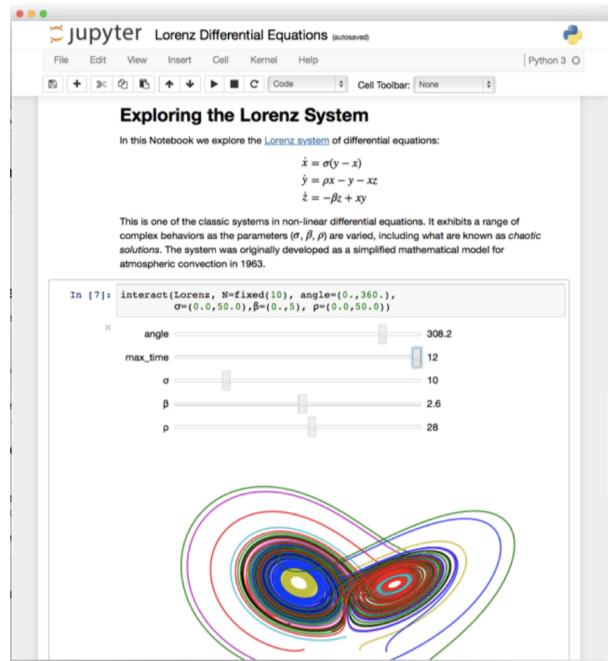
A metodologia empregada para o desenvolvimento de tal algoritmo toma uma abordagem experimental e iterativa. O ponto de partida para a execução do trabalho é a preparação do ambiente tecnológico a ser utilizado. Parte integrante desse ambiente, e primordial ao escopo, são os conjuntos de dados necessários. Esses dados consistem em um conjunto de vídeos que contêm sequências de créditos e cujo momento de início dos créditos é conhecido, bem como o conjunto de metadados respectivos a esses vídeos onde constam os momentos de início dos créditos. Uma vez obtidos esses dados e preparado o ambiente de desenvolvimento, são iniciados os experimentos para obtenção da saída esperada a partir do processamento dos vídeos. Em uma primeira etapa, é adotada uma

postura ingênua frente a resolução do problema, objetivando-se obter algum resultado base com o emprego de técnicas rudimentares. Em um segundo momento, busca-se evoluir a solução utilizando-se de técnicas mais avançadas.

3.1 Ambiente de desenvolvimento

O desenvolvimento do trabalho se apoiou majoritariamente na linguagem de programação Python¹. A escolha da linguagem se deu pela sua simplicidade e grande oferta de bibliotecas que possuem implementações de algoritmos úteis ao trabalho, tais quais bibliotecas para manipulação de imagens e ciência de dados. Outra ferramenta amplamente utilizada foram os notebooks Jupyter², aplicação *web* - acessada através do navegador de internet - de código aberto que permite a execução de códigos em diferentes linguagens de programação, visualização de gráficos e imagens e disposição de conteúdo textual (Figura 8). Sua utilização proporcionou maior agilidade às atividades de exploração necessárias ao desenvolvimento das soluções propostas, uma vez que possibilitou a rápida execução de códigos, visualização de imagens referentes aos *frames* de vídeos e documentação imediata dos resultados parciais obtidos.

Figura 8 - Jupyter Notebook



Fonte: Project Jupyter, 2021

¹ <https://www.python.org/>

² <https://jupyter.org/>

3.2 Obtenção de dados

Para a execução deste trabalho, era imperativa a obtenção de arquivos de vídeo referentes a filmes e séries que contivessem sequências de créditos de encerramento. Igualmente importante era a obtenção dos respectivos valores, em milissegundos, do tempo de início dos créditos em cada um dos vídeos. Os dados utilizados no desenvolvimento do trabalho foram obtidos por meio da disponibilização de acesso em uma plataforma de *streaming*, da qual foi realizado *download* de 100 arquivos de vídeo e de onde foram resgatados os dados contendo os respectivos tempos de início de créditos cadastrados por editores na plataforma.

A plataforma dispunha de uma API (*Application Programming Interface*) através da qual era possível obter informações a respeito de todos os vídeos cadastrados. O acesso à API era feito por meio de requisições HTTP, protocolo de comunicação que permite a troca de dados em uma arquitetura cliente-servidor. Em particular, requisições contendo parâmetros de busca relacionados a metadados de vídeos eram feitas pelo navegador *web* (cliente) enquanto o servidor da aplicação respondia às requisições entregando os recursos solicitados em arquivos JSON³, um formato de troca de informações simples, compacto e legível.

Para realizar-se a escolha dos vídeos a serem baixados e utilizados, foram feitas requisições à API solicitando as informações de todos os vídeos presentes na plataforma. Tais requisições foram feitas utilizando-se a biblioteca de requisições HTTP para Python: Requests⁴. Como resposta às requisições, foram obtidos arquivos JSON com informações de todos os vídeos constantes na plataforma à data, sendo um para vídeos provenientes de séries e outro, de filmes. Para a manipulação desses arquivos, bem como em muitas outras oportunidades ao longo deste trabalho, foi utilizada a biblioteca pandas, que oferece estruturas e operações que ajudam na manipulação de dados. Dessa forma, foi possível estar em posse de uma estrutura tabular de dados na qual cada linha representa um vídeo e cada coluna, uma propriedade do mesmo. Dentre as diversas propriedades disponíveis, destacam-se: o video_id, número inteiro que identifica unicamente o vídeo; o metadado denominado hits.last_day, que indica o número de acessos recebidos pelo vídeo no dia anterior à consulta; e os *cuepoints*, contendo uma lista com todos os pontos de interesse

³ <https://www.json.org/json-en.html>

⁴ <https://docs.python-requests.org/en/master/>

cadastrados no vídeo, com cada ponto sendo representado por uma estrutura que contém um nome e o tempo no vídeo ao qual está associado. Foram selecionados apenas os vídeos nos quais a propriedade *cuepoints* continha um elemento denominado ”*credits_start*” o nome do *cuepoint* que contém o tempo de início dos créditos. Os vídeos selecionados foram, então, ordenados pelo valor decrescente da propriedade *hits_last_day* e os 100 primeiros vídeos resultantes foram os escolhidos para serem utilizados. Dessa forma, foram usados os 100 vídeos, dentre séries e filmes com créditos, mais assistidos no dia anterior à consulta.

Escolhidos os vídeos, o *download* dos mesmos foi feito utilizando-se de outra API fornecida pela plataforma especificamente para o *download* de mídias. As requisições feitas a essa API precisavam indicar o *video_id* e a qualidade de vídeo desejada. Foi desenvolvido um *script* para iterar sobre os vídeos selecionados e fazer a requisição de *download* para cada um. Os vídeos foram baixados na menor qualidade disponível e com as seguintes especificações: vídeos coloridos, em formato mp4, com resolução de 360p e de proporção 16:9, além de uma taxa de 30 *frames* por segundo. Essas características levam à obtenção de 30 imagens (ou frames) RGB por segundo de vídeo, cada imagem sendo formada por 3 componentes com 640 *pixels* de largura contra 360 *pixels* de altura. A manipulação dos arquivos de vídeo foi feita com o auxílio da biblioteca OpenCV⁵, que fornece funções relacionadas à visão computacional e processamento de imagem.

A obtenção dos valores de tempo de início de créditos se deu a partir da mesma tabela de dados gerada anteriormente. Para cada vídeo selecionado, foi resgatado o valor de tempo associado ao elemento *credits_start* da lista de *cuepoints*. Um arquivo ”*cuepoints.csv*” foi gerado contendo 100 linhas, cada uma referente a um vídeo e apresentando, separados por vírgula, seu identificador e o tempo de início dos créditos, conforme cadastrado pelos editores, em milissegundos.

3.3 Modelo de detecção por derivada da diferença entre frames

A primeira abordagem utilizada para a resolução do problema se apoia em uma das principais características observadas nos momentos de transição entre o final do conteúdo e início dos créditos. Em geral, esses momentos são marcados por uma transição abrupta, ilustrada pela Figura 9, entre o último *frame* do conteúdo e o primeiro *frame* dos créditos: enquanto o primeiro ainda costuma apresentar grande quantidade de elementos visuais, implicando em grande variedade de valores de *pixel*, o segundo costuma conter apenas um fundo preto com elementos textuais distribuídos, que implica em *pixels* com menor

⁵ <https://opencv.org/>

variação de valores.

Com base nesse cenário, foi estabelecida a hipótese de que seria possível obter o tempo de início dos créditos no vídeo caso fossem determinados os primeiros *frames* nos quais os valores dos *pixels* apresentassem baixa variação. A validação desta hipótese se deu de forma pragmática com a aplicação de técnicas de processamento de imagens, explanadas ao longo desta seção.

Figura 9 - Transição de fim do conteúdo para o início dos créditos



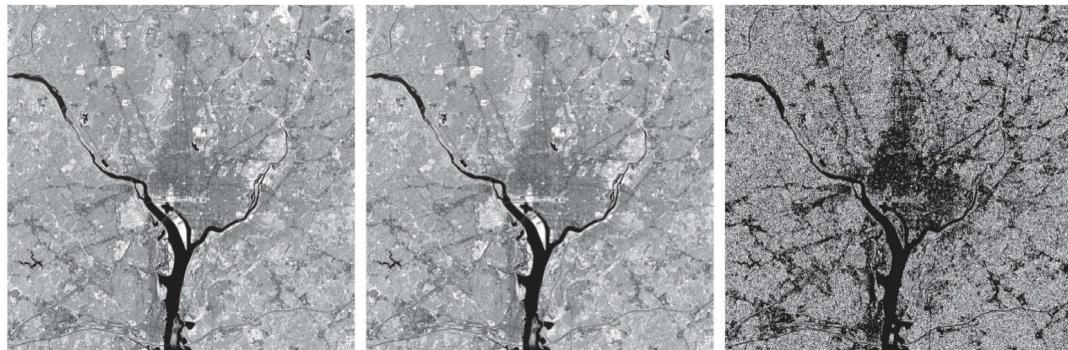
Fonte: Captura de tela realizada pelo autor, 2021

3.3.1 Diferença entre *frames* consecutivos

Considerando-se que uma das características observadas mais relevantes na determinação manual do início dos créditos é baseada na comparação entre *frames* consecutivos, é necessário reproduzir essa comparação de forma procedural. Isso significa fazer a subtração entre os *frames* consecutivos do vídeo. A subtração é usada rotineiramente quando se deseja observar a diferença entre imagens (GONZALEZ; WOODS, 2018). Um exemplo de subtração de imagens é dado na Figura 10.

Da sua representação digital, tem-se que uma imagem pode ser definida como uma matriz de *pixels*, cada qual com um valor de intensidade. Ademais, uma imagem colorida pode ser representada no sistema RGB sendo formada por 3 componentes (matrizes), onde cada uma possui *pixels* com valores indicativos de uma respectiva cor (vermelho, verde ou azul). Dessa forma, a diferença entre duas imagens será dada pela subtração, componente a componente, das matrizes que as representam.

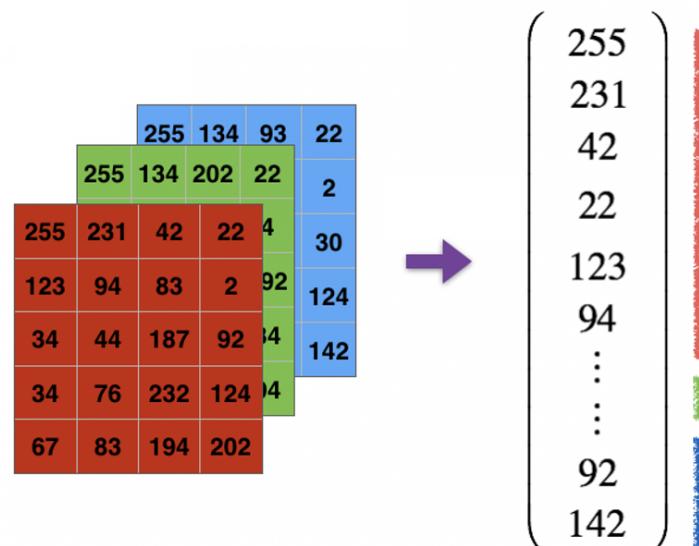
Figura 10 - Subtração de imagens



Terceira imagem resulta da subtração das duas primeiras: cor preta indica que não houve diferença enquanto a cor branca indica que houve variação no valor do pixel. Fonte: GONZALEZ; WOODS, 2018

As matrizes (uma para cada componente de cor) resultantes da subtração indicam, *pixel* a *pixel*, a variação sofrida de um *frame* para o outro. Para fins de simplificação das operações computacionais a serem realizadas, a matriz resultante de dimensões 640x360x3 passa a ser considerada como um vetor cuja dimensão equivale ao produto entre as dimensões originais da matriz, ou seja, 691200. Essa operação é ilustrada na Figura 11. Os valores *pixel* a *pixel* obtidos, entretanto, não proporcionam uma medida facilmente comparável de quanto grande ou pequena foi a variação entre os frames. Uma tal medida pode ser obtida do módulo, ou norma euclidiana, do vetor resultante. Com esse valor, é possível realizar uma comparação direta da diferença entre quaisquer *frames* consecutivos do vídeo.

Figura 11 - Reorganização de matrizes em um vetor



Fonte: Adaptação. BALSYS, 2021

3.3.2 Derivada da diferença entre *frames* consecutivos

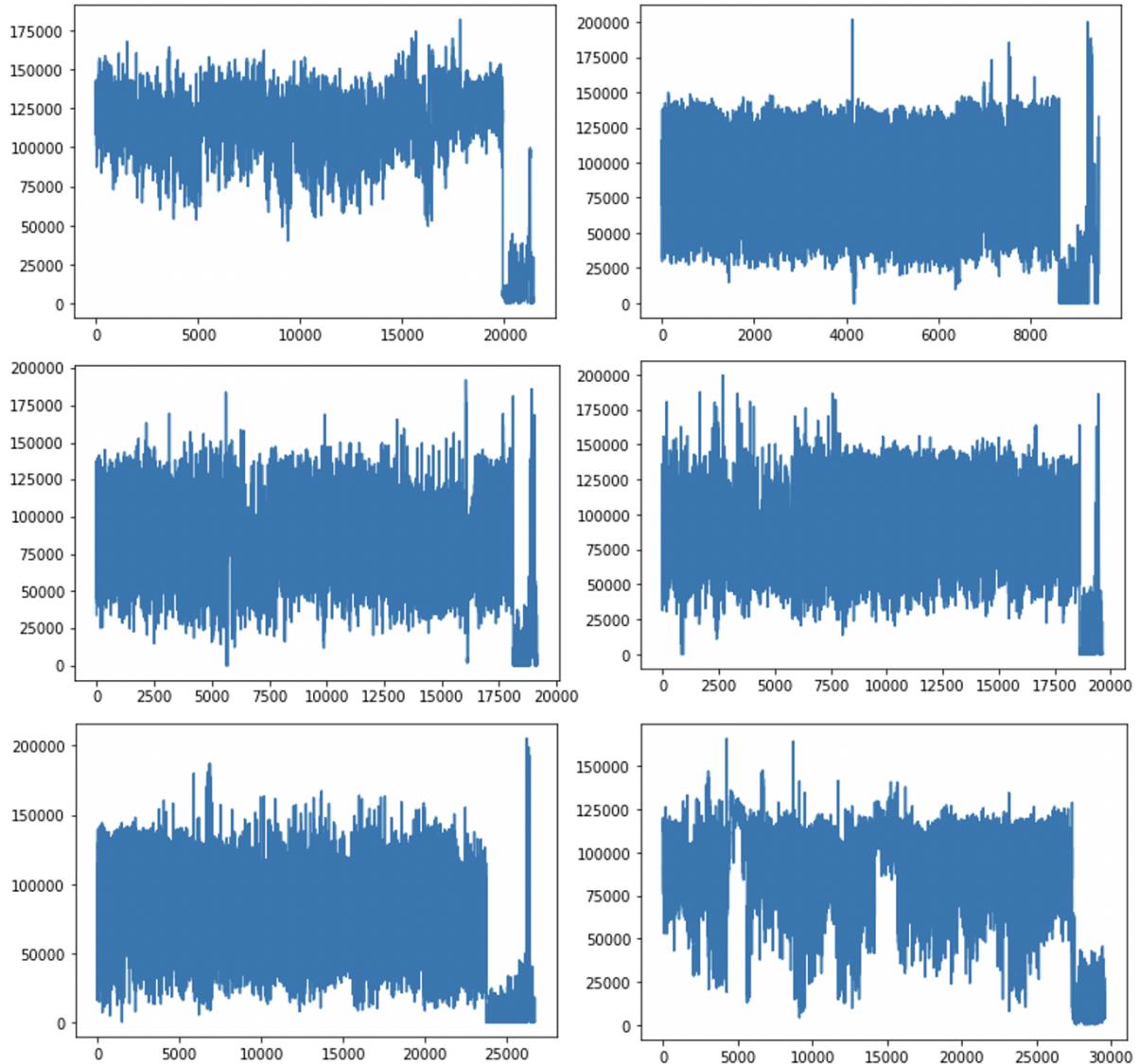
Como um vídeo consiste em uma sequência de imagens, pode ser interpretado como uma série temporal. Desse modo, cada imagem pode ser associada a um momento t no tempo, com t um número inteiro. Uma visão global no escopo do vídeo no que diz respeito ao módulo da variação entre *frames* consecutivos pode ser dada por uma função que associa um valor a cada diferença entre *frames* consecutivos no tempo, conforme 1. Para um vídeo com n frames, existirão $n-1$ valores que indicam a variação entre um par de *frames* consecutivos.

$$X = x_1, x_2, \dots, x_n \text{ com } n \in \mathbb{I} \text{ e } S_t \text{ o } frame \text{ do vídeo no momento } t; \\ X(t) = \|S_{t+1} - S_t\|, \text{ para } 0 \leq t < n \quad (1)$$

A obtenção do momento de início dos créditos, então, será dependente da obtenção do momento em que se inicia o decrescimento da variação entre os valores dos *pixels* dos *frames* consecutivos. Tal decrescimento pode ser avaliado através da investigação da taxa de variação dos valores da função. Isto é, para melhor avaliar o momento de início dos créditos, é preciso conhecer a derivada da função em 1. Dessa forma, foi obtida a derivada através do método numérico de diferenças finitas. O resultado foi avaliado em termos de valor absoluto e pode ser observado na Figura 12.

A investigação visual dos valores obtidos já é capaz de sugerir que a hipótese inicial é verdadeira. Quando os valores são dispostos em um gráfico de linha é possível perceber que existe um momento bem definido a partir do qual os valores de variação entre as imagens são substancialmente menores.

Figura 12 - Derivada da normas das diferenças entre *frames*



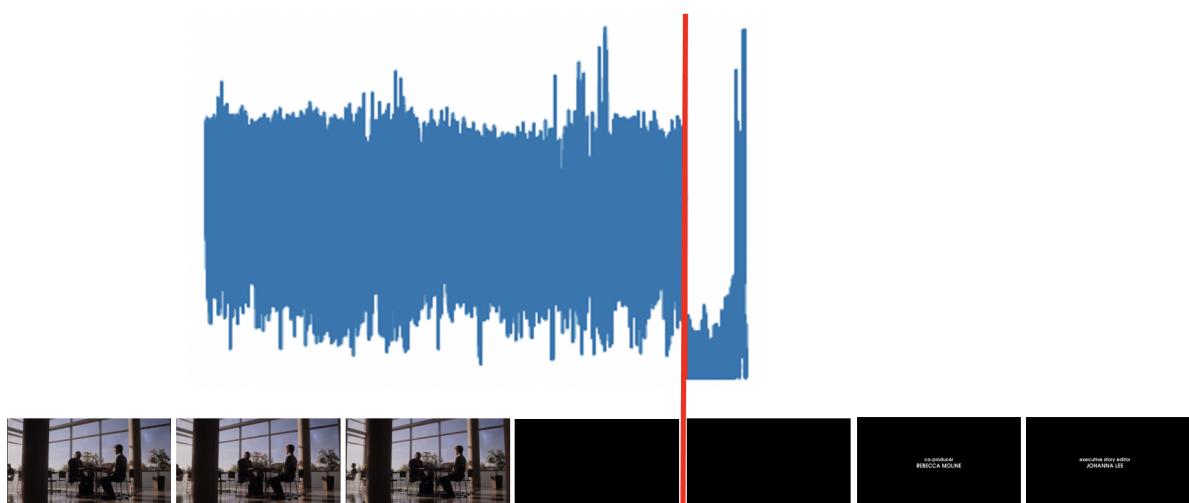
Gráficos de linha mostrando a taxa de variação da norma da diferença entre *frames* (eixo vertical) para diferentes vídeos. Fonte: O autor, 2021

3.3.3 Obtenção do tempo de início dos créditos

Uma análise gráfica dos resultados obtidos, explorada na Figura 13, evidencia a existência de um segmento na série de valores o qual apresenta números menores. Esse segmento coincide com o momento final do vídeo e pode, justamente, ser interpretado como sendo referente à sequência de créditos. Para a determinação do ponto inicial desse segmento, inicialmente precisa ser estabelecido um tamanho mínimo para considerá-lo válido. Na prática, essa determinação significa escolher o tempo mínimo necessário com

baixa variação dos *frames* para que um segmento seja considerado como créditos. Esta etapa é importante para evitar que outras sequências do vídeo, como transições e cenas mais lentas ou estáticas, sejam equivocadamente interpretadas como créditos. Assim, partindo dos $n-2$ valores obtidos da derivada das normas das diferenças entre frames, com n igual ao número de *frames* do vídeo, é aplicada uma média móvel com janela referente ao tamanho mínimo aceitável. Finalmente, o ponto que apresenta menor média é o escolhido como ponto de início dos créditos. O índice deste ponto na sequência de valores indica a respectiva ordem do *frame* correspondente no vídeo. Para a obtenção do tempo em milisegundos, basta fazer a divisão do número do *frame* pela taxa de *frames* por segundo do vídeo e a conversão de unidade.

Figura 13 - Identificação gráfica do momento de início dos créditos



Momento de início dos créditos identificado, indicado pela linha vermelha. Fonte: O autor, 2021

Apesar de apresentar resultados promissores, a solução proposta também falhou em experimentos com alguns vídeos. Notadamente, quando os créditos não atendiam os padrões previamente estabelecidos ou quando o vídeo continha outras sequências com baixa variação das imagens. A partir deste momento, é explorada outra abordagem para a resolução do problema.

3.4 Modelo de detecção por classificador de frames

Almejando evoluir a solução proposta, alternativas ao modelo apresentado anteriormente foram investigadas. Considerando-se que existem alguns padrões determinantes na identificação manual do ponto de interesse, tal qual a taxa de variação da diferença entre frames, a próxima abordagem propõe a utilização de Inteligência Artificial para o

reconhecimento desses padrões e, consequentemente, detecção do início dos créditos.

3.4.1 Inteligência Artificial

A Inteligência Artificial (IA) é o ramo da computação que se dedica a encontrar maneiras de se obter de computadores ou recursos computacionais comportamentos que remetam à inteligência humana. Segundo Schalkoff (1990), “é o campo de estudo que tenta explicar e simular o comportamento inteligente em termos de processos computacionais”. Já Kurzweil (1992) define como “a arte de criar máquinas que executam funções que requerem inteligência quando executadas por pessoas”. Chamados agentes inteligentes, produtos dessa área de estudo são sistemas capazes de perceber e atuar em um ambiente, sendo autônomos, adaptáveis a mudanças e atuando com objetivos (NORVIG; RUSSEL, 2013).

3.4.2 Aprendizado de máquina

Dentre as áreas abrangidas pela IA, figura a de Aprendizado de Máquina. Esta área é focada no estudo de algoritmos que se aperfeiçoam automaticamente por meio de experiência e uso de dados (MITCHELL, 1997). Em geral, algoritmos de aprendizado de máquina são agentes que percebem um determinado conjunto de características (dados de entrada) com base no qual atuam tomando algum tipo de decisão (dados de saída).

Típicos problemas de aprendizado de máquina são a classificação e a regressão. Problemas de classificação são aqueles nos quais busca-se atribuir um rótulo, ou classe, a uma determinada entrada de acordo com suas características. Problemas de regressão são aqueles nos quais deseja-se realizar uma predição a partir de outras observações.

Os algoritmos dessa área de estudo podem ser divididos em três categorias: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. No aprendizado supervisionado, o algoritmo precisa ser treinado com um conjunto de dados de entrada para os quais os dados de saída já são conhecidos, sendo capaz de generalizar o conhecimento e aplicar em exemplos desconhecidos. No aprendizado não supervisionado, o algoritmo não tem acesso aos dados de saída esperados, sendo responsável por identificar padrões nos dados de entrada. Em aprendizado por reforço, o agente “aprende” sendo recompensado por acertos e penalizado por erros.

O problema tratado neste trabalho pode se aproveitar do potencial dos algoritmos de aprendizado de máquina, uma vez que existem dados suficientes e um objetivo bem definido.

3.4.3 Solução proposta

A ideia é criar um agente inteligente, doravante referido como modelo, capaz de reproduzir os pensamentos e ações envolvidos no racional da pessoa editora no seu processo de trabalho. Em especial, um modelo capaz de decidir, assim como o editor, se um dado *frame* pertence ou não a uma sequência de créditos. Com um modelo capaz de classificar todos os *frames* de um vídeo como pertencentes ou não à sequência de créditos, é possível determinar o tempo de início da sequência a partir da localização, no vídeo, dos primeiros *frames* classificados positivamente.

Formulando o problema em termos de aprendizado de máquina, trata-se de um problema de classificação binária - onde existem duas classes de saída possíveis para o algoritmo. As classes de saída possíveis são: “pertence à sequência de créditos”, também referida apenas como “créditos” e “não pertence à sequência de créditos”, também referida apenas como “não-créditos”. Ademais, trata-se de um problema de aprendizado supervisionado, uma vez que o algoritmo será treinado utilizando-se das informações (quais *frames* pertencem ou não à sequência de créditos) advindas do processo de detecção humana.

3.4.4 Dados de treino

O conjunto de dados obtido contém, além dos *cuepoints* de início, os *cuepoints* de fim dos créditos. Dessa forma, considerando-se o segmento do vídeo que vai do início ao fim dos créditos, é possível mapear todos os *frames* pertencentes à sequência de encerramento. Nesse contexto, amostras de *frames* da classe créditos podem ser extraídas desse segmento, enquanto amostras de *frames* da classe não-créditos podem ser extraídas do restante do vídeo.

Foram geradas amostras de *frames* pertencentes a ambas as classes para cada um dos 100 vídeos disponíveis. Foram extraídos 3 *frames* por segundo de vídeo, ao longo de toda a duração do mesmo. Um arquivo csv foi gerado para cada vídeo, com uma linha para cada frame, discriminando se o *frame* pertencia ou não à sequência de créditos através de um valor booleano ”true” ou ”false”. Isso levou à obtenção de milhares de amostras por vídeo, variando o número de acordo com seu tempo de duração, e centenas de milhares de amostras ao todo, das quais a maioria era referente a exemplos negativos, ou não-créditos.

3.4.5 Seleção dos parâmetros de entrada

Uma das etapas mais importantes dentro de um processo de aprendizado de máquina é a seleção dos parâmetros de entrada, ou *features*, do modelo. É a partir dos valores assumidos por esses parâmetros em cada amostra que o algoritmo de aprendizado de máquina será capaz de identificar padrões nos dados e estabelecer um método de classificação adequado.

Uma escolha trivial de parâmetro de entrada para o problema seria o uso dos próprios frames. Nesse caso, a entrada seria uma imagem de 640x360 *pixels* com 3 componentes de cor. Para fins de simplificação, como apresentado na Figura 11, a imagem é reorganizada em um vetor de dimensão 691200. Ou seja, utilizar diretamente a imagem como entrada levaria o modelo a necessitar de 691200 *features*. Esse número poderia ser atenuado reduzindo-se a dimensão da imagem mas ainda assim levaria a um número elevado de parâmetros, que não necessariamente é sinônimo de um modelo melhor.

Descartando-se a opção trivial, são investigadas formas de se aproveitar o conhecimento humano a respeito de sequências de créditos de forma a produzir boas *features* para um modelo. Primeiramente, sabe-se que os créditos costumam estar situados nos minutos finais do vídeo, o que significa que existe uma relação direta entre o início dos créditos e a ordem dos *frames* na série temporal determinada pelo vídeo. É sabido também que *frames* pertencentes a créditos frequentemente possuem conteúdo textual. Também é notório que esse conteúdo textual pode apresentar um movimento característico, locomovendo-se para regiões distintas da imagem ou mesmo realizando um movimento vertical (as populares letras "subindo"). Além disso, é novamente explorada a diferença entre *frames* consecutivos.

3.4.5.1 Ordem do frame

A primeira *feature* estabelecida é dada pelo número que define a posição do *frame* na sequência de imagens que compõem o vídeo. *Frames* que pertencem à sequência de créditos têm esse número maior do que os que não pertencem. Essa *feature* foi obtida diretamente da leitura do vídeo, que é feita *frame* a *frame* e na ordem em que eles aparecem. Foi denominada "frame_nb".

3.4.5.2 Norma da diferença entre *frames*

Característica já explorada anteriormente, a norma da diferença entre *frames* consecutivos indica o quanto os *pixels* da imagem variaram de um quadro para o outro.

Sequências de créditos costumam apresentar baixas variações. Essa *feature* foi denotada por “pixel_norm_diff”.

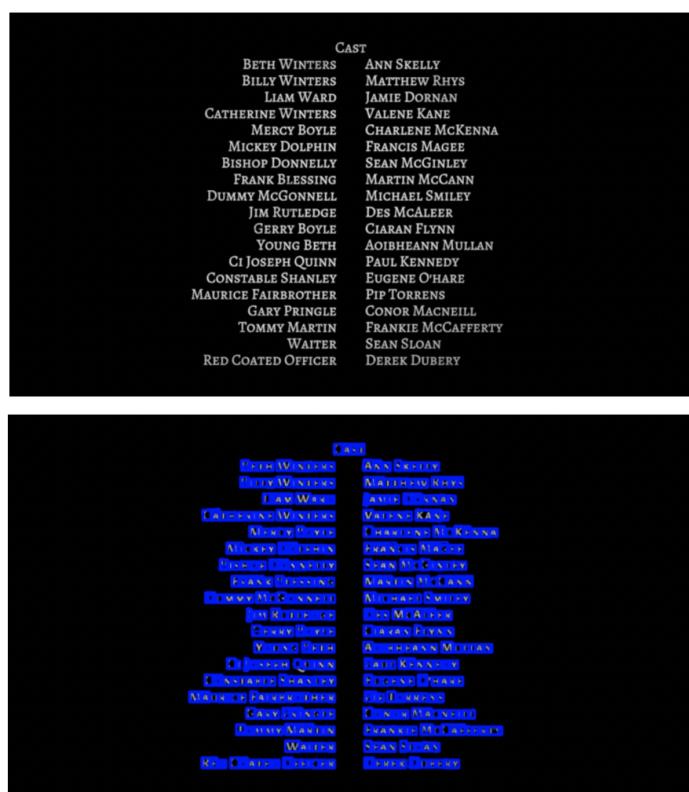
3.4.5.3 Norma da diferença entre histogramas

Feature com proposta semelhante à anterior, porém avalia a norma da diferença entre os histogramas de *frames* consecutivos. Denotada por “hist_norm_diff”.

3.4.5.4 Número de letras

A primeira *feature* abordando o aspecto do conteúdo textual presente nas imagens. Consiste no número de regiões retangulares que delimitam letras na imagem. Esse parâmetro é obtido do *frame* utilizando-se uma técnica de visão computacional para a detecção de regiões na imagem, através do algoritmo MSER (maximally stable extremal regions) (MATAS O. CHUM; PAJDLA, 2002). Exemplos das regiões detectadas pelo algoritmo podem ser vistos na Figura 14. Essa *feature* é referida como “nb_rectangles”.

Figura 14 - Segmentação de regiões da imagem que contêm letras



Em azul, retângulos que delimitam letras na imagem. Fonte: O autor, 2021

3.4.5.5 Fluxo óptico

Por fim, para abordar o movimento característico das letras durante a sequência de créditos, utiliza-se o conceito de fluxo óptico. O fluxo óptico é a distribuição do movimento aparente do padrão de brilho de uma imagem (HORN; SCHUNCK, 1981). Trata-se do campo vetorial que indica a movimentação ocorrida em *frames* consecutivos. Foi utilizado o método diferencial de Lucas–Kanade para estimativação de fluxo óptico, em implementação do opencv. A partir dos resultados, três *features* foram extraídas:

- x_flow: valor médio da componente horizontal do fluxo
- y_flow: valor médio da componente vertical do fluxo
- nb_flow_points: número de pontos em que houve movimento

Um fator importante a se considerar é o caráter temporal do vídeo: *frames* dos créditos estarão agrupados no tempo. Assim, faz sentido considerar como *features*, para um determinado *frame* parâmetros referentes à *frames* posteriores ao atual. Dessa forma, o conjunto de parâmetros possibilita a identificação de comportamentos que perdurem na sequência de frames, como é o caso dos créditos. Foi estabelecida uma janela de 10 *frames* na geração dos parâmetros, de forma que cada amostra possuirá 70 *features* (as 7 *features* descritas anteriormente para o *frame* atual e para os 9 seguintes).

Todos os 100 vídeos foram processados e, *frame* a *frame*, foram obtidos os valores dos parâmetros. Esses valores foram salvos nos respectivos arquivos csv criados anteriormente contendo a classificação dos *frames* de cada vídeo.

3.4.6 Pré-processamento dos dados

Antes de se fornecer os dados de entrada ao modelo para o treinamento, é importante atentar-se à qualidade desses dados. Dados de má qualidade podem levar a criação de um modelo enviesado e que não corresponda à realidade. Processos comuns de garantia de qualidade de dados são a correção de valores de parâmetros incorretos ou faltosos, por exemplo. Problemas dessa sorte são comumente encontrados no desenvolvimento de modelos cujos dados são provenientes de cadastros, *websites* ou outras fontes não confiáveis. Neste trabalho, entretanto, os dados de treino foram fabricados exclusivamente para uso no modelo, de forma que cada parâmetro tem seu valor preenchido corretamente.

Porém, outros tipos de pré-processamento se fazem necessários. Primeiramente, como cada *feature* é referente a uma métrica diferente extraída da imagem, naturalmente possuem valores em escalas diferentes. É preciso que esses valores sejam normalizados,

de forma a evitar que valores que estejam em escalas maiores tenham maior importância no modelo do que aqueles que estejam em escalas menores. A normalização dos dados foi feita com uso da classe Normalizer do pacote de pré-processamento do scikit-learn.

Além da normalização dos dados, uma tarefa importante a ser feita antes de se dar início ao treinamento de um modelo é o rebalanceamento do conjunto de dados. Isso porque, do conjunto disponível, existe uma quantidade maior de exemplos negativos do que positivos (mais *frames* não-créditos do que *frames* créditos). Esse desequilíbrio poderia levar a um modelo com maior tendência a classificar um *frame* como não-créditos. Possíveis soluções para esse problema são a subamostragem e a sobreamostragem. Tais soluções consistem, respectivamente, na retirada de amostras da classe majoritária e adição de amostras à classe minoritária. Como o número de amostras obtido foi suficientemente grande, optou-se pela subamostragem como forma de equilibrar o conjunto de dados.

3.4.7 Classificador

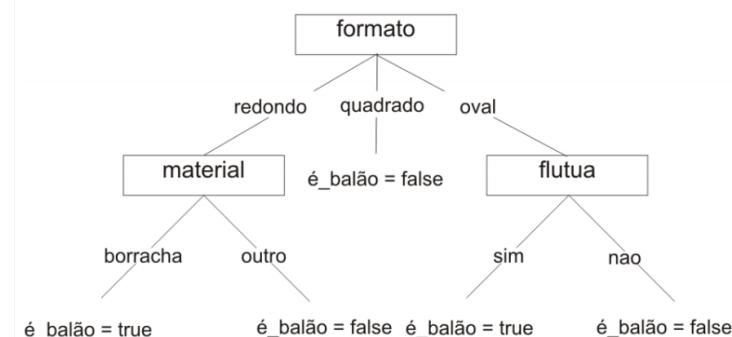
Uma vez determinados os parâmetros de entrada e realizado o pré-processamento necessário, pode-se dar início ao desenvolvimento do modelo de classificação de *frames*. São avaliados dois algoritmos distintos, os quais são alimentados com os mesmos dados e submetidos às mesmas etapas de avaliação. O conjunto de dados foi dividido de forma que 77% das amostras foram destinadas ao treinamento dos algoritmos, enquanto os 23% restantes foram reservados exclusivamente para o teste dos modelos. Foram utilizadas implementações dos algoritmos de aprendizado de máquina disponíveis na biblioteca scikit-learn⁶. Os resultados apresentados pelos modelos são descritos no Capítulo 4

3.4.7.1 Floresta Aleatória

Uma classe de algoritmos utilizados na resolução de problemas de classificação são as árvores de decisão. As árvores de decisão figuram entre os mais populares algoritmos de aprendizado de máquina, graças a sua simplicidade e fácil entendimento (WU et al., 2008). Em suma, esse algoritmo busca criar uma estrutura de decisão na qual os parâmetros de entradas de maior importância são representados como nós mais próximos a raiz da árvore, enquanto as folhas representam a classificação decorrente de cada ramo. Um exemplo de árvore de decisão pode ser observado na Figura 15.

⁶ <https://scikit-learn.org/stable/>

Figura 15 - Exemplo de árvore de decisão



Árvore de decisão para classificar se um objeto é ou não um balão. Fonte: ARAUJO, 2004

Uma técnica, denominada *ensemble*, propõe a combinação de múltiplos algoritmos de aprendizado de máquina para obtenção de melhor performance (ROKACH, 2010). O método de *ensemble* pode ser utilizado com árvores de decisão, originando as chamadas florestas aleatórias. O modelo de floresta aleatória consiste em ajustar um determinado número de árvores de decisão em diferentes subconjuntos dos dados de treino. Seus resultados são dados em função da média dos resultados das árvores treinadas. O modelo foi treinado usando 100 árvores de decisão.

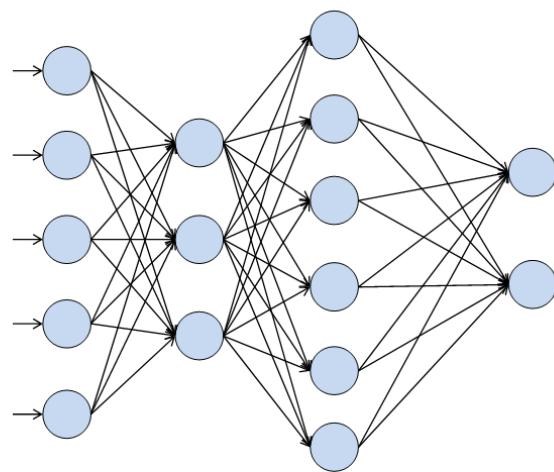
3.4.7.2 Redes Neurais

A segunda classe de algoritmos de aprendizado de máquina utilizada para a criação do classificador foi a das redes neurais artificiais. Este algoritmo tem como inspiração as redes de neurônios do cérebro. Ele pode ser representado por uma estrutura em grafo, onde cada nó é análogo ao neurônio e a transição entre os nós representam as sinapses. Uma arquitetura de rede popular é a de perceptron multicamada, onde os nós estão distribuídos entre uma camada de entrada, uma de saída e camadas intermediárias, chamadas escondidas. Na Figura 16, um exemplo de rede multicamada.

Treinar um modelo de rede neural significa ajustar pesos associados a cada transição entre nós, avaliando o erro entre a saída esperada e a obtida após a propagação dos dados de entrada pela rede. O algoritmo que costuma ser utilizado para realizar este ajuste é o de retro-propagação (RUMELHART; HINTON; WILLIAMS, 1986).

A rede neural treinada para este trabalho contou com 70 nós na camada de entrada, referentes a cada um dos parâmetros, uma camada escondida com 100 nós e a camada de saída com um nó para cada classe (créditos e não-créditos).

Figura 16 - Exemplo de rede neural multicamadas



Fonte: LIMA, 2016

4 RESULTADOS

Neste capítulo são detalhados os resultados obtidos ao longo do desenvolvimento das soluções propostas. Ao início de cada sessão, são definidas as métricas de avaliação utilizadas. Em um primeiro momento, são discutidos os resultados oriundos do modelo de detecção por derivada da diferença entre *frames*. Em seguida, são dispostos e comparados os resultados dos classificadores treinados e o seu desempenho na identificação do momento de início dos créditos.

4.1 Diferença entre *frames*

Como já evidenciado através de análise gráfica (Figuras 12 e 13), a diferença entre *frames* consecutivos é capaz de indicar um segmento do vídeo que possui menor variação entre as imagens e ao qual potencialmente pertence o início da sequência de créditos. Se faz necessária, entretanto, a avaliação numérica dos resultados. A avaliação deste método se deu pela comparação entre os valores de início de crédito resultantes da técnica com aqueles oriundos da detecção manual. Uma métrica denominada “erro” foi criada e definida conforme 2:

$$\text{erro} = \text{tempo_obtido_manualmente} - \text{tempo_obtido_automaticamente} \quad (2)$$

O valor da métrica de erro pode ser: igual a 0 quando os tempos são iguais, indicando 100% de precisão; maior que 0 quando o tempo obtido pela solução é menor, indicando que o início dos créditos detectado está adiantado em relação ao tempo cadastrado manualmente; menor que 0 quando o tempo obtido pela solução é maior, indicando que o início dos créditos detectado está atrasado em relação ao tempo cadastrado manualmente.

Como o valor do tempo de início dos créditos foi obtido a partir de uma média móvel sobre a série de valores de variação entre *frames*, diferentes cenários foram avaliados variando-se o tamanho da janela da média móvel. Foram avaliadas janelas de 1, 3, 5 e 10 segundos. Estatísticas descritivas dos resultados para a métrica de erro, para cada cenário, são apresentadas na Tabela 1.

Tabela 1 - Estatísticas descritivas do erro da técnica de diferença entre *frames*

	1s	3s	5s	10s
média	87.86	977.81	-13.22	-18.96
desvio padrão	304.38	570.00	175.34	177.67
mínimo	-683.74	-286.15	-682.77	-682.77
primeiro quartil	-4.18	621.12	-29.89	-27.08
mediana	1.58	781.90	-1.86	-0.64
terceiro quartil	79.23	1318.14	2.43	2.31
máximo	1416.98	3260.69	788.89	776.04

Unidade de medida: segundos. Fonte: O autor, 2021

A análise dos dados mostra que o uso de janelas de média móvel menores levam à maiores valores médios da métrica de erro, indicando que nesses casos o tempo de início dos créditos estava, em média, adiantado em relação ao tempo marcado manualmente. Por outro lado, as janelas de 5 e 10 segundos levaram a erros médios negativos, ou seja, os tempos de início de créditos obtidos usando esses parâmetros estavam, em média, atrasados em relação a marcação manual. Além disso, vale observar os valores medianos. Um erro de -1,86 segundos significa que a funcionalidade de “pular créditos” será disponibilizada com um atraso de cerca de 2 segundos. Em contrapartida, um erro de 781,9 segundos implica na apresentação da funcionalidade cerca de 13 minutos antes do esperado. Entende-se que tempos adiantados tem impacto negativo maior do que tempos atrasados, uma vez que podem levar o usuário a deixar prematuramente o conteúdo.

Por fim, considerando-se o possível impacto negativo da apresentação prematura da funcionalidade de ‘pular créditos’, foi estabelecido uma tolerância para o erro de até 3 segundos. A Tabela 2 mostra a quantidade de vídeos, dentre os 100, nos quais o tempo de início de créditos detectado estava correto, atrasado ou, no máximo, até 3 segundos adiantado.

Tabela 2 - Total de marcações aceitáveis

1s	3s	5s	10s
57	37	80	82

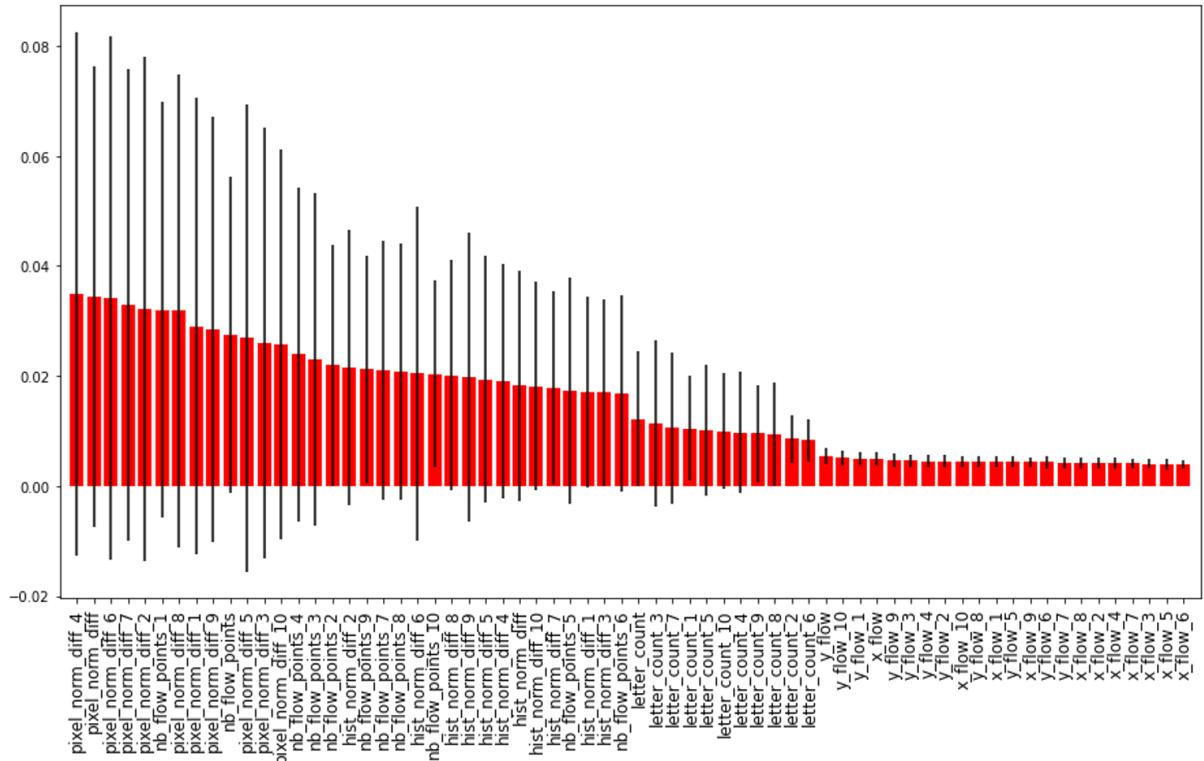
Fonte: O autor, 2021

4.2 Classificação de *frames*

Aqui são apresentados os resultados referentes à utilização dos modelos de aprendizado de máquina baseados nos algoritmos de floresta aleatória e redes neurais. Para fins de comparação, os modelos foram treinados primeiramente utilizando-se de dados desbalanceados, com número de amostras de classe não-créditos muito superior as de classe créditos. Em seguida, novos modelos foram treinados usando conjunto de dados balanceado por meio de subamostragem.

Após o treinamento dos modelos de floresta aleatória, foi possível observar a relevância de cada uma das 70 *features* usadas para a classificação. A Figura 17 mostra o grau de importância relativo de cada uma. É possível perceber que os parâmetros relacionados à variação entre *frames* consecutivos (*pixel_norm_diff* e *hist_norm_diff*) e o parâmetro relacionado ao número de pontos de movimento (*nb_flow_points*) são os que apresentam maior grau de importância para o modelo. Em contrapartida, os parâmetros que menos contribuíram foram os valores médios das componentes do fluxo óptico (*x_flow* e *y_flow*).

Figura 17 - Importância das *features* para o modelo de floresta aleatória



Fonte: O autor 2021

A partir de agora, são apresentados os resultados de performance dos 4 modelos treinados. A avaliação dos modelos se deu por meio da utilização de métricas típicas em

problemas de classificação, as quais são definidas a seguir.

Um instrumento utilizado na avaliação de modelos de classificação é a matriz de confusão. Ela consiste em uma tabela que apresenta os números de verdadeiros positivos (classificações corretas da classe créditos), falsos negativos ou erro tipo II (modelo prevê não-créditos para uma amostra da classe créditos), falsos positivos ou erro tipo I (modelo prevê créditos para uma amostra da classe não-créditos e os verdadeiros negativos (classificações corretas da classe não-créditos) (RODRIGUES, 2019).

Da matriz de confusão aplicada ao problema em questão, derivam-se as métricas de:

- acurácia: quantidade de classificações corretas sobre o total de classificações;
- precisão: quantidade de classificações corretas dentre as classificações do tipo créditos;
- *recall*: quantidade de amostras do tipo crédito corretamente classificadas como créditos;
- *F1-score*: média harmônica entre precisão e *recall*;

A avaliação dos modelos se deu em duas etapas. Em um primeiro momento, foi utilizado todo o conjunto de dados de teste, e em um segundo momento, foram realizados testes de validação cruzada, onde o conjunto de dados de teste é dividido em subconjuntos sobre os quais são avaliados os modelos (KOHAVI, 1995). A validação cruzada ocorreu com a divisão dos dados em 5 conjuntos.

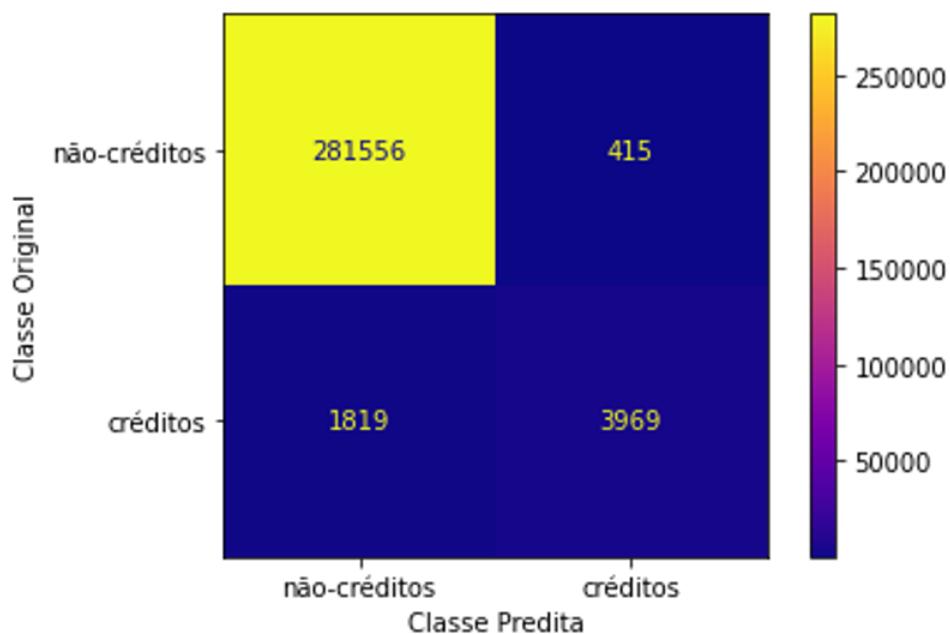
As matrizes de confusão resultantes de cada modelo são apresentadas nas Figuras 18, 19, 20 e 21 e os resultados da avaliação com o conjunto de teste completo são apresentados na Tabela 3. As tabelas 4, 5, 6, 7 contém as métricas obtidas na validação cruzada.

Tabela 3 - Métricas de acurácia e *F1-score* dos modelos

	modelo 1	modelo 2	modelo 3	modelo 4
acurácia	0.99	0.94	0.98	0.90
<i>F1-score</i>	0.78	0.94	0.69	0.90

Modelo 1: floresta aleatória desbalanceada; Modelo 2: floresta aleatória balanceada; Modelo 3: rede neural desbalanceada; Modelo 4: rede neural balanceada. Fonte: O autor, 2021

Figura 18 - Matriz de confusão da floresta aleatória com dados desbalanceados



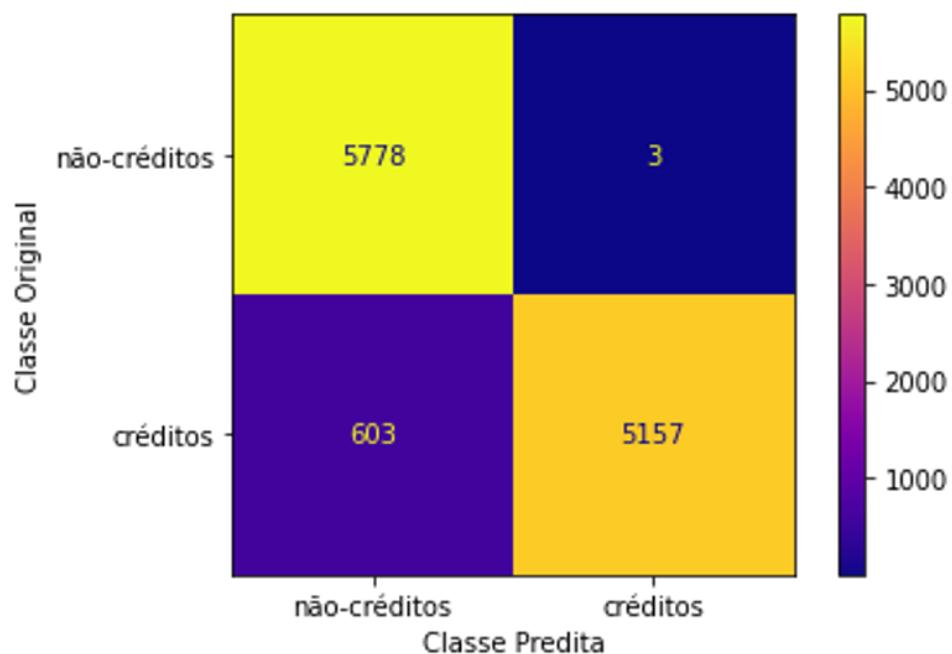
Fonte: O autor 2021

Tabela 4 - Métricas de desempenho da floresta aleatória com dados desbalanceados

	precisão	recall	duração do treino	duração da classificação
1	0.94	0.93	2438.29	5.03
2	0.93	0.75	2527.15	5.12
3	0.91	0.79	2471.75	4.91
4	0.93	0.81	2271.88	3.90
5	0.93	0.72	2430.58	4.09
média	0.92	0.8	2427.93	4.61

Fonte: O autor, 2021.

Figura 19 - Matriz de confusão da floresta aleatória com dados balanceados



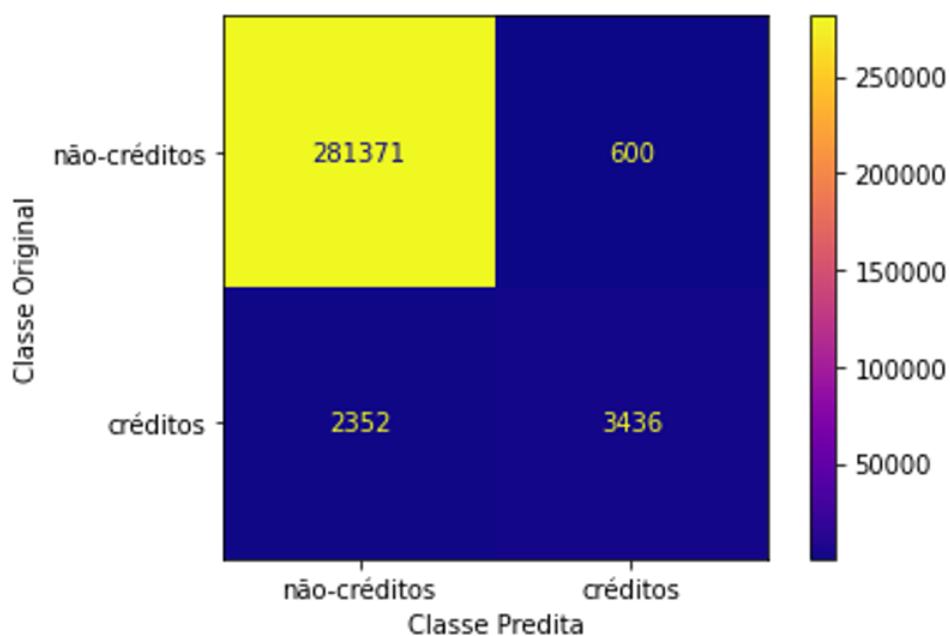
Fonte: O autor 2021

Tabela 5 - Métricas de desempenho da floresta aleatória com dados balanceados

	precisão	<i>recall</i>	duração do treino	duração da classificação
1	0.96	0.96	28.41	0.87
2	0.86	0.85	47.71	0.13
3	0.92	0.92	26.06	0.10
4	0.89	0.89	20.55	0.15
5	0.88	0.88	24.87	0.10
média	0.91	0.9	31.52	0.25

Fonte: O autor, 2021.

Figura 20 - Matriz de confusão da rede neural com dados desbalanceados



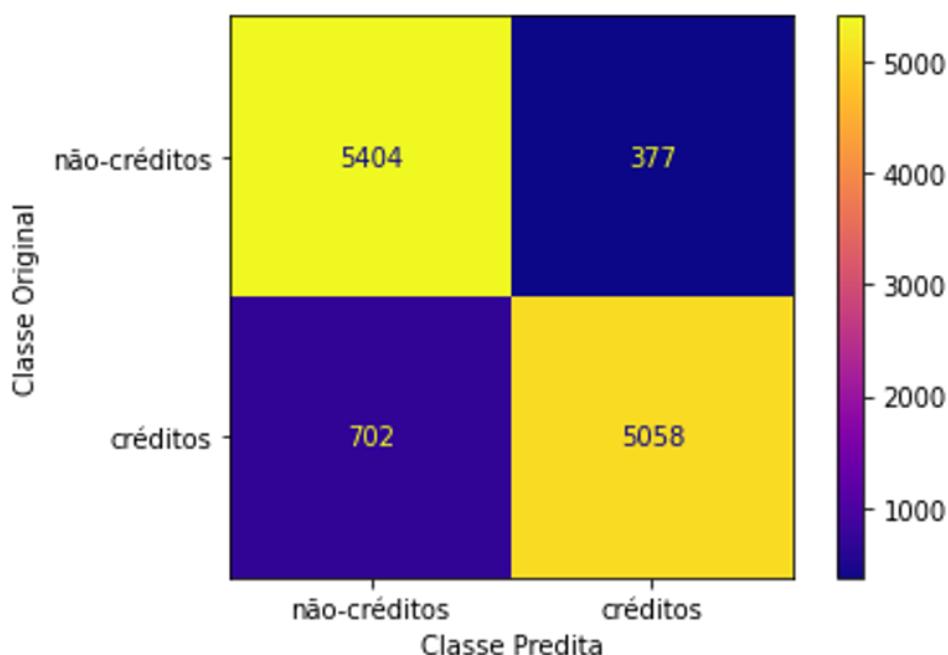
Fonte: O autor 2021

Tabela 6 - Métricas de desempenho da rede neural com dados desbalanceados

	precisão	recall	duração do treino	duração da classificação
1	0.93	0.89	537.77	6.07
2	0.87	0.74	613.72	5.79
3	0.91	0.78	1384.52	4.75
4	0.92	0.74	1068.57	5.36
5	0.92	0.72	850.74	4.53
média	0.91	0.77	891.06	5.3

Fonte: O autor, 2021.

Figura 21 - Matriz de confusão da floresta aleatória com dados balanceados



Fonte: O autor 2021

Tabela 7 - Métricas de desempenho da rede neural com dados balanceados

	precisão	<i>recall</i>	duração do treino	duração da classificação
1	0.95	0.95	50.53	0.09
2	0.89	0.88	51.80	0.09
3	0.89	0.89	58.31	0.10
4	0.87	0.87	57.94	0.09
5	0.87	0.86	72.19	0.12
média	0.89	0.89	58.16	0.1

Fonte: O autor, 2021.

Da análise das matrizes de confusão e das métricas de avaliação dos modelos, é possível notar que os modelos treinados com dados desbalanceados performaram melhor em termos de acurácia e precisão. Entretanto, os valores de *recall* foram consideravelmente maiores nos modelos que utilizaram dados平衡ados. Ou seja, nos modelos em que a distribuição da quantidade de amostras de classes positiva e negativa foi equilibrada, classificações corretas de amostras da classe positiva (créditos) foi mais frequente.

4.3 Detecção de créditos usando classificador

Finalmente, são analisados os resultados associados aos tempos de início de créditos obtidos da utilização dos classificadores de *frames*. Esse tempo é obtido de forma análoga ao método da Sessão 4.1. As figuras 22, 23, 24 e 25 apresentam gráficos de linha contendo a classificação dada pelos modelos para os *frames*, a probabilidade do *frame* ser classificado pelo modelo como créditos e o momento de início de créditos obtido para diversos vídeos. Essas visualizações reforçam os resultados apontados na seção anterior quanto às métricas de precisão e *recall* dos classificadores. É possível perceber que os modelos balanceados apresentam maior probabilidade de rotular um *frame* como créditos ao longo de toda a duração do vídeo.

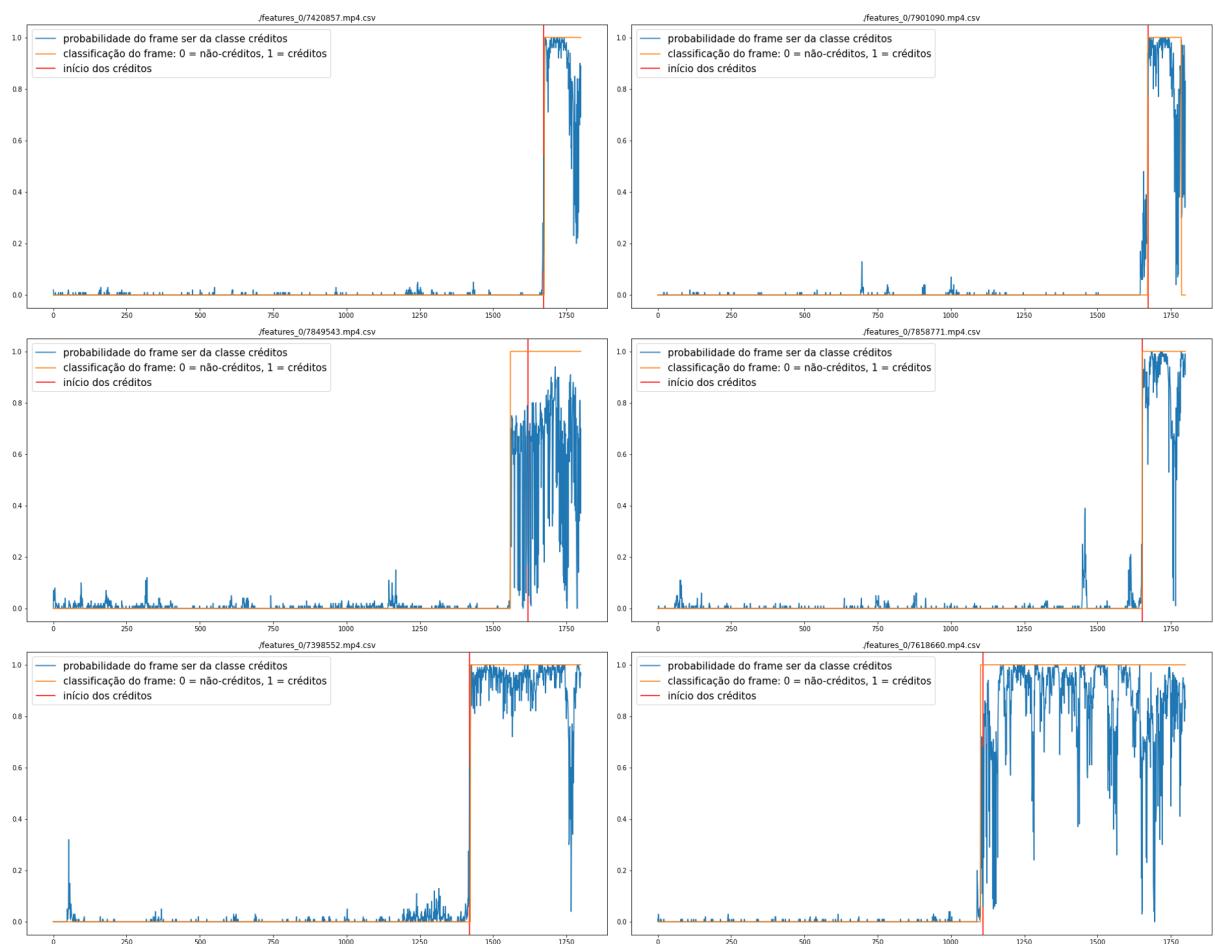
Uma comparação entre os resultados dos tempos de início de créditos detectados com base nas classificações de cada um dos modelos é feita aproveitando-se a métrica de erro definida na seção 4.1 e é apresentada na Tabela 8. Observa-se que o uso de modelos não balanceados leva a um erro médio menor. Ademais, a faixa de valores assumidos pelos erros das detecções baseadas nos modelos desbalanceados se inicia em valores mais altos e cresce mais rapidamente, o que significa que os tempos de início de créditos obtidos a partir desses modelos estão mais frequentemente adiantados em relação ao tempo marcado editorialmente. Nesse contexto, o método de detecção mais eficaz é o que utiliza o modelo de floresta aleatória com amostra desbalanceada. Essa técnica foi capaz de identificar corretamente o momento de inicio de créditos de 86% de um conjunto de vídeos desconhecidos, com tolerância de 3 segundos para mais ou para menos.

Tabela 8 - Estatísticas descritivas do erro nos modelos de classificadores de *frames*

	modelo 1	modelo 2	modelo 3	modelo 4
média	12.05	32.23	7.34	25.95
desvio padrão	75.41	102.37	175.34	96.78
mínimo	-34.75	-8.75	-205.44	-57.08
primeiro quartil	-0.90	2.80	-1.47	2.06
mediana	0.11	5.01	0.58	5.02
terceiro quartil	0.62	7.72	1.72	8.15
máximo	559.33	561.00	558.66	561.33

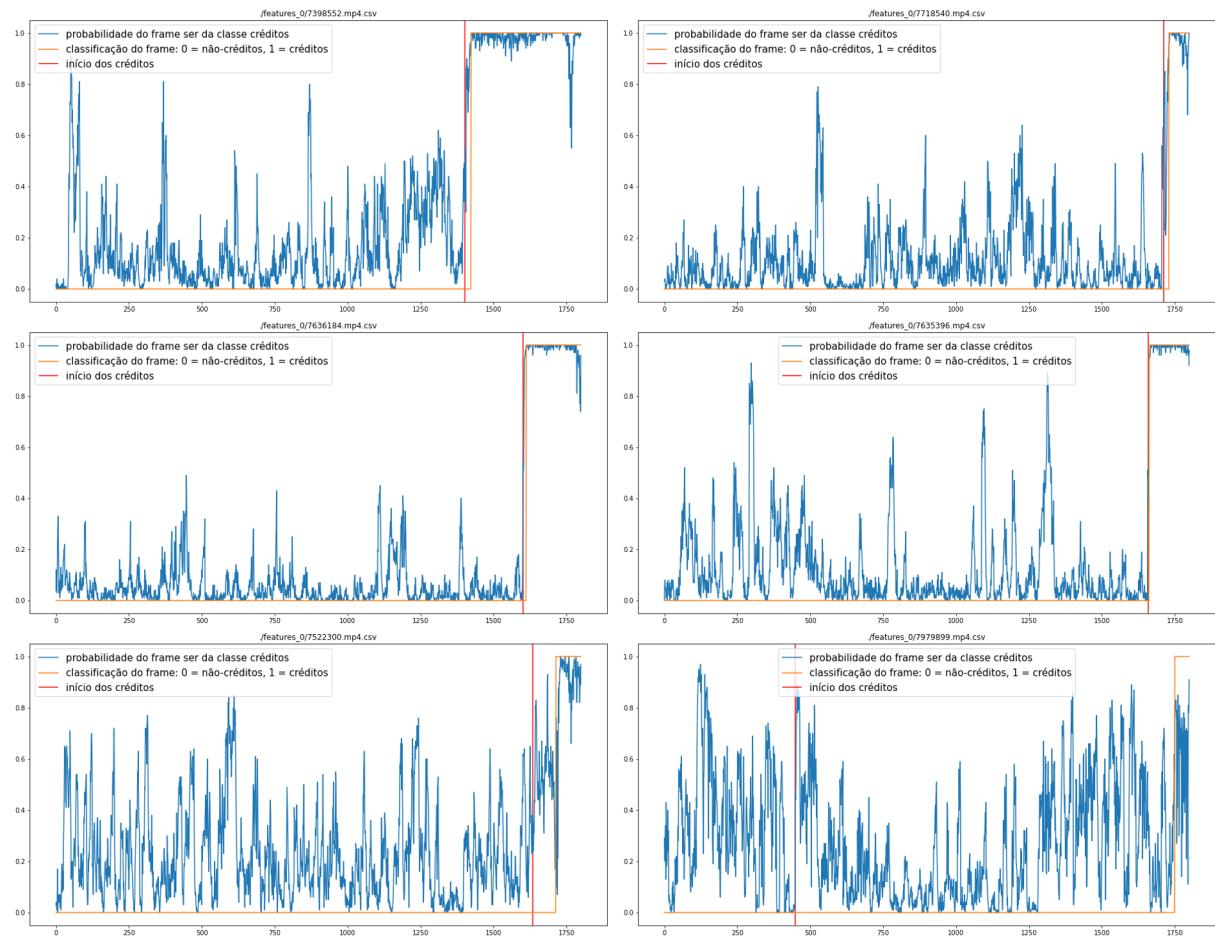
Modelo 1: floresta aleatória desbalanceada; Modelo 2: floresta aleatória balanceada; Modelo 3: rede neural desbalanceada; Modelo 4: rede neural balanceada. Fonte: O autor, 2021

Figura 22 - Análise gráfica da detecção usando floresta aleatória com dados desbalanceados



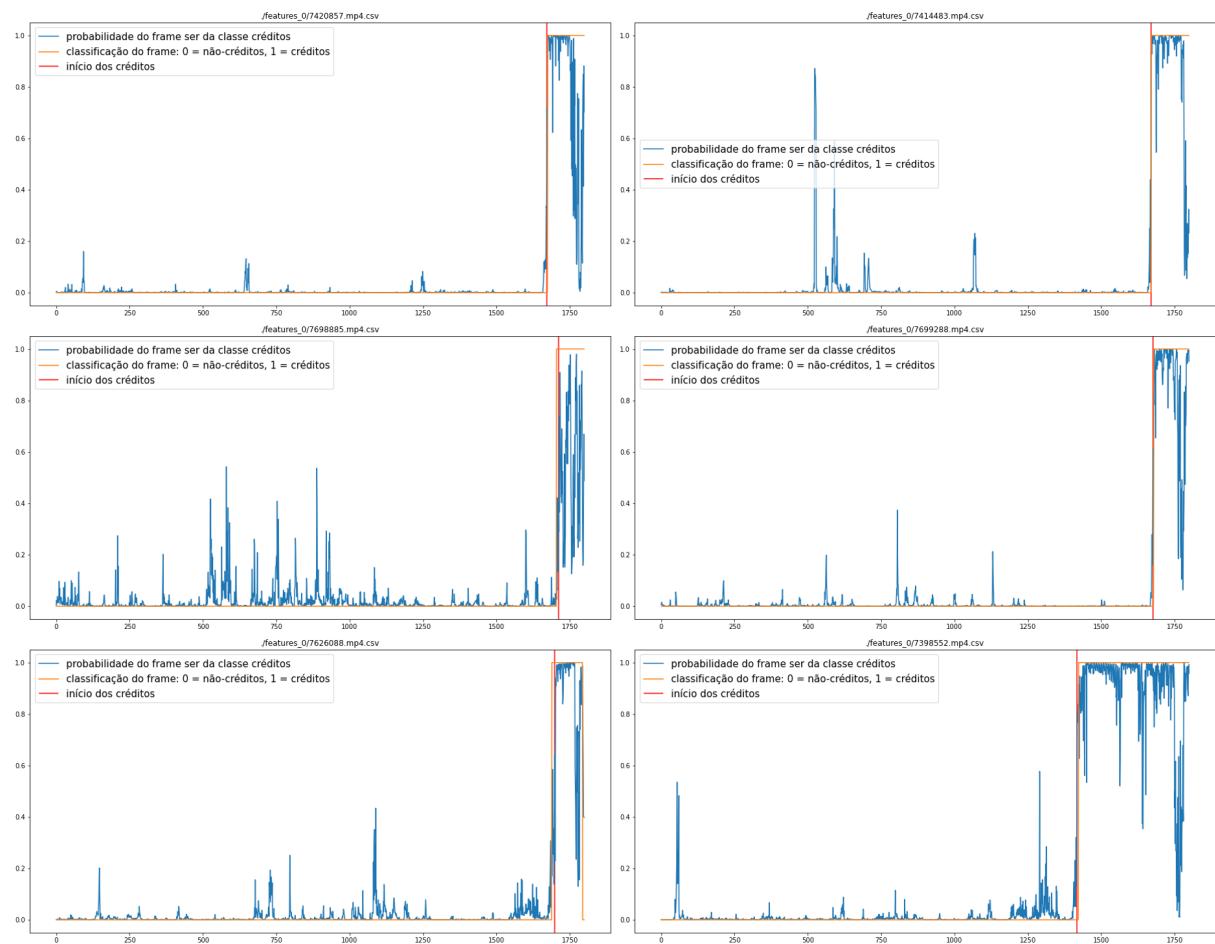
Fonte: O autor 2021

Figura 23 - Análise gráfica da detecção usando floresta aleatória com dados balanceados



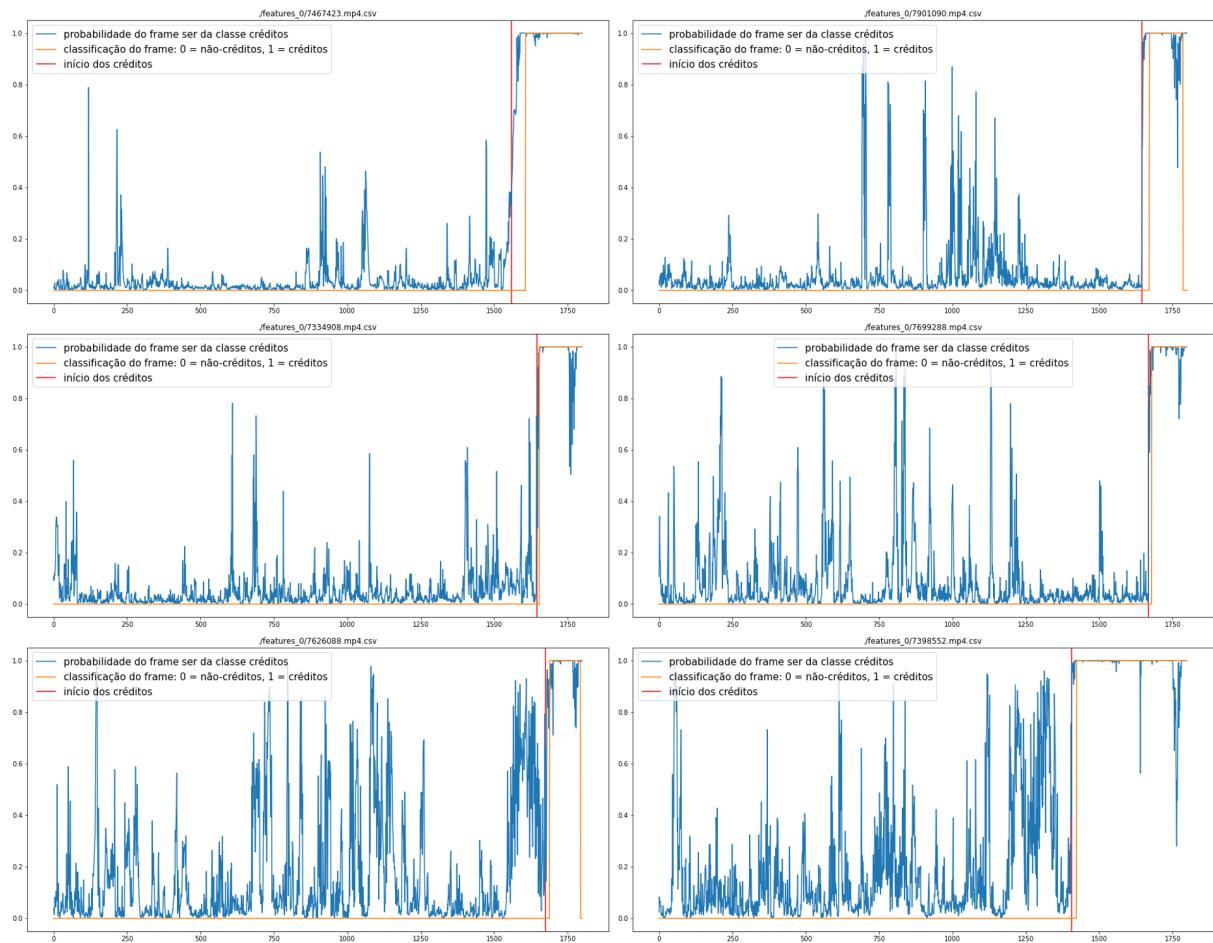
Fonte: O autor 2021

Figura 24 - Análise gráfica da detecção usando redes neural com dados desbalanceados



Fonte: O autor 2021

Figura 25 - Análise gráfica da detecção usando rede neural com dados balanceados



Fonte: O autor 2021

4.4 Pesquisa Reproduzível

Todo o material desenvolvido para este trabalho foi disponibilizado no GitHub através do projeto “End Credits Detection”, podendo ser acessado em <https://github.com/thiago-franco/end-credits-detection>.

Foram disponibilizados os modelos treinados, o *software* desenvolvido para realizar a detecção do momento de início dos créditos dado um arquivo de vídeo e todos os dados utilizados para o treinamento dos modelos.

CONCLUSÃO

Após uma introdução do tema a ser tratado, este texto apresentou uma detalhada contextualização no que se refere ao ecossistema de plataformas de *streaming* de vídeos. Foram explorados os principais elementos que compõem a arquitetura de tais plataformas, com destaque para aqueles pertencentes ao fluxo de consumo direto do vídeo pelo usuário. A relação entre os metadados de vídeo e as funcionalidades que um *player* de vídeos pode prover foi estabelecida para, assim, descrever-se o problema em questão: a detecção manual do tempo de início de créditos de encerramento em produtos audiovisuais. Delimitado o escopo do problema, foram propostas soluções com diferentes abordagens, as quais foram desenvolvidas e tiveram seus resultados expostos.

Baseada na avaliação da derivada da norma da diferença entre cada par de *frames* consecutivos de um vídeo, a primeira solução proposta mostrou-se, a partir de investigação gráfica, aderente à hipótese que levou a sua proposição: o momento de início dos créditos é marcado por uma queda acentuada da variação dos valores dos *pixels* das imagens. Esta característica, entretanto, não foi suficiente para detectar com precisão o tempo de início dos créditos. Esta solução se apoiou no uso de uma média móvel para encontrar o momento no qual a variação entre os *frames* iniciava seu decrescimento. Através dos resultados obtidos, foi possível perceber que essa abordagem levava a tempos com erro significativo, traduzindo-se em marcações atrasadas ou adiantadas demais em relação à marcação editorial.

Com o intuito de se obter melhores resultados, foram estudadas novas abordagens para a resolução do problema. Foi empregado o uso de inteligência artificial por meio de algoritmos de aprendizado de máquina para a criação de modelos capazes de classificar se um dado *frame* pertence ou não à uma sequência de créditos. Com todos os *frames* de um vídeo classificados, a detecção do tempo de início dos créditos se deu a partir da identificação do primeiro de um grupo de *frames* que passam a ser classificados como créditos. Os modelos treinados se basearam em florestas aleatórias e redes neurais. Ademais, foram analisados os resultados do treinamento dos modelos usando conjuntos de dados balanceados e desbalanceados. Apesar do balanceamento dos dados de treino ser uma prática que pode levar a melhores resultados, notou-se que ela não é adequada para o problema em questão. Isso porque os vídeos possuem, por natureza, um volume muito superior de *frames* que não pertencem à sequência de créditos e, ao prover um modelo com uma quantidade equilibrada de amostras positivas e negativas, acaba-se por favorecer a classificação das amostras como exemplos positivos, ou seja, como pertencentes à sequência de créditos. Com isso, mais imagens tendem a ser classificadas como créditos, mesmo aquelas que não pertencem ao encerramento, elevando a possibilidade de erro e diminuindo a precisão do método. Dentre os métodos que utilizaram o conjunto des-

balanceado de dados, a floresta aleatória mostrou-se uma melhor opção de classificador, apresentando precisão superior à da rede neural.

Com a avaliação dos resultados dos métodos desenvolvidos, é possível concluir que foi elaborada uma solução eficaz, que cumpre com os requisitos propostos e, inclusive, supera expectativas. Os melhores resultados levaram a uma taxa de 86% de acerto do tempo de início dos créditos, com uma margem de erro de 3 segundos para mais ou para menos. Quando permitido um atraso ligeiramente maior do tempo detectado, de 5 segundos, esse número pode superar 90%. Além disso, algumas sequências de crédito que não estavam previstas no escopo de atuação deste trabalho, como aquelas contendo algum conteúdo dinâmico, foram corretamente detectadas. Assim sendo, este trabalho deixa como contribuição um modelo de detecção de início de créditos em vídeos de séries e filmes, que pode ser usado tanto para a oferta de funcionalidades como a de pular créditos quanto em quaisquer outras oportunidades que necessitem deste metadado do vídeo. Contribui também com o estudo documentado da metodologia empregada e todo o aprendizado decorrente.

Como trabalho futuro, existe muito espaço para a melhoria dos modelos de classificação propostos. Podem ser exploradas ajustes de hiperparâmetros bem como diferentes arquiteturas de redes, por exemplo. O emprego de redes neurais convolucionais possivelmente pode extrair melhor o padrão de características das imagens, levando a resultados potencialmente superiores. Além disso, podem ser investigadas outras características das sequências de créditos para a geração de *features* adicionais.

REFERÊNCIAS

- AMERI, Mina; HONKA, Elisabeth; XIE, Ying. Viewing modus and media franchise engagement. SSRN, 2019. Disponível em: <<http://dx.doi.org/10.2139/ssrn.2986395>>.
- ARAUJO, Ricardo Matsumura de. Aprendizado de máquina em sistemas complexos multiagentes: estudo de caso em um ambiente sob racionalidade limitada. 2004. Disponível em: <https://www.researchgate.net/publication/242695289_Aprendizado_de_maquina_em_sistemas_complexos_multiagentes_estudo_de_caso_em_um_ambiente_sob_racionalidade_limitada>.
- BALSYS, Rokas. *Understanding Logistic Regression*. 2021. Disponível em: <<https://pylessons.com/Logistic-Regression-part2/>>. Acesso em: 23 Jun. 2021.
- CISCO. *Cisco Predicts More IP Traffic in the Next Five Years Than in the History of the Internet*. 2018. Disponível em: <<https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1955935>>.
- DHANANI, Suhel; PARKER, Michael. *Digital Video Processing for Engineers*. 1. ed. [S.l.]: Elsevier, 2012.
- GONZALEZ, Rafael C.; WOODS, Richard E. *Digital Image Processing*. 4. ed. 330 Hudson Street, New York, NY 10013: Pearson, 2018.
- HORN, Berthold K.P.; SCHUNCK, Brian G. Determining optical flow. 1981. Disponível em: <https://www.researchgate.net/publication/222450615_Determining_Optical_Flow>.
- KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. 1995.
- KURZWEIL, Raymond. *The Age of Intelligent Machines*. [S.l.]: MIT Press, 1992.
- LIMA, Edirlei Soares de. *Redes Neurais*. 2016. Disponível em: <https://edirlei.com/aulas/ia_2016_2/IA_Aula_14_Redes_Neurais_2017.html>. Acesso em: 6 mai. 2021.
- MATAS O. CHUM, M. Urban J.; PAJDLA, T. Robust wide baseline stereo from maximally stable extremal regions. *Proc. of British Machine Vision Conference, pages 384-396*, 2002.
- MITCHELL, Tom M. *Machine Learning*. 1. ed. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997.
- NORVIG, Peter; RUSSEL, Stuart. *Inteligência Artificial*. 3. ed. [S.l.]: Pearson, 2013.
- Project Jupyter. *The Jupyter Notebook*. 2021. Disponível em: <<https://jupyter.org/>>. Acesso em: 23 Jul. 2021.
- RODRIGUES, Vitor. *Métricas de Avaliação*. 2019. Disponível em: <[https://vitorborbarodrigues.medium.com/m% C3% A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c](https://vitorborbarodrigues.medium.com/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c)>. Acesso em: 26 Jun. 2021.

- ROKACH, L. Ensemble-based classifiers. *Artificial Intelligence Review*, 2010. Disponível em: <<https://link.springer.com/article/10.1007/s10462-009-9124-7>>.
- SCHALKOFF, Robert J. *Artificial Intelligence: An Engineering Approach*. [S.l.]: McGraw-Hill College, 1990.
- STEINER, Emil. Binge-watching motivates change: Uses and gratifications of streaming video viewers challenge traditional tv research. *Convergence: The International Journal of Research into New Media Technologies*, 2018. Disponível em: <<http://david.choffnes.com/classes/cs4700sp14/papers/akamai.pdf>>.
- WU, Xindong et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 2008.
- WU Y. T. HOU, Wenwu Zhu Ya-Qin Zhang Dapeng; PEHA, J. M. Streaming video over the internet: approaches and directions. *IEEE Transactions on Circuits and Systems for Video Technology*, 2001.