UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM

FACULDADE DE TECNOLOGIA - FT

BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Extração de características de narrativas audiovisuais a partir de elementos visuais e suas relações

Elton Dione Nascimento de Alencar

Manaus - AM

2022

Elton Dione Nascimento de Alencar

Extração de características de narrativas audiovisuais a partir de elementos visuais e suas relações

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia da Universidade Federal do Amazonas como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia da Computação.

Orientador

Prof. Edjard de Souza Mota, PhD.

Universidade Federal do Amazonas - UFAM

Faculdade de Tecnologia - FT

Manaus - AM

2022

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

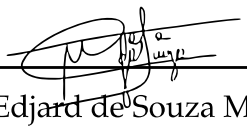
Alencar, Elton Dione Nascimento de
A368e Extração de características de narrativas audiovisuais a partir de elementos visuais e suas relações / Elton Dione Nascimento de Alencar . 2022
61 f.: 31 cm.

Orientador: Edjard de Souza Mota
TCC de Graduação (Engenharia da Computação) - Universidade Federal do Amazonas.

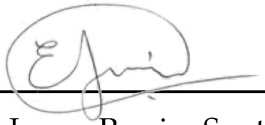
1. Detecção de objetos. 2. Detecção de ação. 3. Composição de cena. 4. Modelos pré-treinados. 5. Elementos de narrativas audiovisuais (vídeos). I. Mota, Edjard de Souza. II. Universidade Federal do Amazonas III. Título

Monografia de Graduação sob o título *Extração de características de narrativas audiovisuais a partir de elementos visuais e suas relações*, apresentada por Elton Dione Nascimento de Alencar, submetida ao corpo docente do curso de Engenharia da Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários para a obtenção do grau de engenheiro.

Aprovado por:



Prof. Edjard de Souza Mota, Ph.D.
Orientador
Instituto de Computação
Universidade Federal do Amazonas



Eduardo James Pereira Souto, Dr.
Instituto de Computação
Universidade Federal do Amazonas

Prof. José Luiz de Souza Pio, Dr.
Instituto de Computação
Universidade Federal do Amazonas

Manaus - AM, 17 de junho de 2022.

Dedico este trabalho à todos os amigos, colegas, professores e profissionais da UFAM que contribuíram para o meu crescimento profissional, durante minha jornada na graduação.

Agradecimentos

Em primeiro lugar, agradeço à minha mãe, Gilmara, por todo o esforço e garra dedicados, desde sempre, para que eu chegasse até aqui. Não foi fácil e sem ela não seria possível. Em seguida, agradeço aos meus irmãos, minhas tias e, principalmente, minha avó Dircinha (*in memoriam*), que infelizmente, não pode estar presente nesta etapa tão importante da minha vida. Obrigado, vocês são o motor disso tudo.

Aos amigos da faculdade que tive a honra de conquistar durante a minha jornada na graduação, os quais contribuíram, diretamente, nesse período. Em especial à Ariel Luane, Larissa Pessoa, Mario Hirotoshi, Pedro Matias e Victória Guimarães, pelo tempo, risadas, experiências e, até mesmo, dificuldades que compartilhamos.

Ao meu orientador e aos colegas do grupo de pesquisa científica em Inteligência Artificial, que ajudaram no desenvolvimento desta pesquisa, em especial à Cris Cirino, pesquisadora e doutoranda da Universidade Federal do Para, a qual tem um papel importante neste trabalho. À Universidade Federal do Amazonas, por toda estrutura e oportunidades que tenho a honra de poder me beneficiar. A todos os professores, por todo conhecimento compartilhado comigo até aqui.

No geral, agradeço a todos que me ajudaram e contribuíram, de uma forma ou de outra, para que tudo isso fosse possível. Até porque, sozinho a gente não chega em lugar nenhum.

We can only see a short distance ahead, but we can see plenty there that needs to be done.

-Alan Turing

Extração de características de narrativas audiovisuais a partir de elementos visuais e suas relações

Autor: Elton Dione Nascimento de Alencar

Orientador: Prof. Edjard de Souza Mota, PhD.

Resumo

Os vídeos compartilhados na internet representam mais da metade de todo o tráfego mundial. O YouTube foi considerada a segunda plataforma de compartilhamento de vídeos mais utilizada atualmente. Portanto, a otimização dos processos de análise do conteúdo presente nessas estruturas audiovisuais, é considerada uma solução para tornar esses processos mais efetivos. Sendo assim, este trabalho apresenta o desenvolvimento de uma ferramenta, em Python, que tem como objetivo automatizar o processo de extração de características visuais de vídeos publicados no YouTube. Para isso, foi validada uma arquitetura com os principais processos executados na ferramenta proposta, a qual faz a aplicação de dois modelos de *deep-learning*, pré-treinados, um para detecção de objetos e o outro para detecção de ações/movimentos presentes no vídeo. A ferramenta proposta pode ser integrada no processo de caracterização e análise de narrativas audiovisuais, uma vez que, de acordo com os resultados obtidos, as características extraídas pela ferramenta se relacionam com os principais elementos que compõem os recursos visuais da narrativa.

Palavras-chave: Detecção de Objetos, Detecção de ação, Composição de Cena, Modelos pré-treinados, Elementos de Narrativas Audiovisuais, Vídeos, YouTube.

Extração de características de narrativas audiovisuais a partir de elementos visuais e suas relações

Autor: Elton Dione Nascimento de Alencar

Orientador: Prof. Edjard de Souza Mota, PhD.

Abstract

Videos shared on the internet represent more than 80% of the traffic worldwide. YouTube is considered the second most used platform for video sharing. The optimization of the processes related to content analysis regarding these audiovisual structures is treated as a solution to make these processes more effective. For these reasons, this work presents a tool, developed in Python, which aims to automate the process of extracting visual features from videos shared on YouTube. For this, an architecture with the main processes executed in the proposed tool, which applies two pre-trained deep-lerning models, one for object detection and the other for action detection, was validated. Based on the result obtained, this approach can be integrated into the analysis process of audiovisual narratives, since the characteristics extracted by the tool are related to the main elements that compose the visual resources of the narrative.

Keywords: Object Detection, Action Detection, Scene Description, Video Understanding, Audiovisual Narrative, YouTube.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

1.1 Considerações iniciais

Foi estimado que, em 2022, os vídeos compartilhados na internet representarão mais de 80% de todo o tráfego mundial (CISCO, 2020). Portanto, com a grande quantidade de vídeos disponíveis diariamente na internet, o desenvolvimento de ferramentas que automatize a identificação e extração de características de vídeos é um trabalho necessário que otimizaria parte do processo de análise do conteúdo de cenas presente nessas estruturas audiovisuais (CIRINO et al., 2021).

Para realização da análise de narrativas audiovisuais, inicialmente, é necessário a extração de características (elementos) que compõem as cenas do vídeo em estudo. Pois, a análise da relação desses elementos ao longo do mesmo contribui para o entendimento da mensagem criada para transmitir informações, opiniões, ideias, sensações e sentimentos, a partir dos seus recursos audiovisuais (CIRINO, 2021).

A linguagem audiovisual é composta por recursos visuais que expressam parte da narrativa presente em um vídeo. Onde, ainda segundo o que afirma Cirino (2021), a composição de cenas, movimentação da câmera, entre outros, são elementos que compõem esses recursos. Elementos esses que constituem as imagens (*frames*), que são compostas pelos objetos, presentes no ambiente/cenário, que se relacionam ao longo do vídeo.

Sendo assim, durante a análise da narrativa audiovisual, a identificação e a extração de objetos, presentes em um *frame*, fazem parte dos processos usados para compreensão do conteúdo e descrição de uma cena, a partir da análise das relações desses objetos com os outros elementos que compõem os recursos visuais, presentes no vídeo.

1.2 Motivação

A quantidade de informações presentes em um vídeo é tanta que há, na literatura, trabalhos que defendem a utilização de técnicas computacionais no processo de identificação e extração de características que compõem esse vídeo (FAN et al., 2021). Além disso, sabe-se que para a análise de narrativas audiovisuais, o analista, inicialmente, gasta um esforço para realizar o processo de identificação e extração das características visuais presentes nos vídeos (CIRINO et al., 2021).

O desenvolvimento de uma ferramenta que faça a identificação e extração automática de parte dessas características, utilizando técnicas computacionais, é uma solução que otimizaria o processo de análise e caracterização das narrativas audiovisuais. Sendo esta a principal motivação para esta proposta.

Portanto, este trabalho propõe uma ferramenta, desenvolvida em Python (MIHAJLOVIĆ et al., 2020), que faz a extração automática de características visuais, através da utilização de dois modelos de *deep-learning*, pré-treinados. Um para detecção de objetos (BOCHKOVSKIY; WANG; LIAO, 2020) e o outro para detecção de ações/movimentos (FEICHTENHOFER et al., 2019), presentes em um vídeo. Sendo esta solução possível de se tornar integrável no processo de análise de narrativas audiovisuais.

A validação desta proposta é feita através da relação das características extraídas pela ferramenta desenvolvida, com as características que compõem uma narrativa audiovisual, levando em consideração o que Cirino (2021), Doutoranda no Programa de Pós-Graduação em comunicação na Universidade Federal do Pará, afirma sobre os elementos da narrativa audiovisual, os quais podem ser caracterizados pelos recursos visuais presentes no vídeo.

O fato de a ferramenta proposta ser desenvolvida através da utilização de soluções e técnicas computacionais, no seu estado da arte, além do uso da API do YouTube, como fonte dos dados utilizados para os experimentos, caracteriza este trabalho como uma prova de conceito da aplicação dos modelos utilizados, combinados com a manipulação de ferramentas públicas desenvolvidas em Python. Tudo isso para otimizar o processo de análise e, consequentemente, entendimento de vídeos.

Além disso, vale ressaltar que este trabalho está inserido em um contexto de

pesquisa mais amplo, que originou-se a partir de uma parceria entre o Laboratório de Inteligência Artificial do PPGI/ICOMP/UFAM, e uma pesquisa de doutorado do Programa de Pós-Graduação em comunicação, na Universidade Federal do Pará. Pesquisa essa que tem com título *"A Desinformação sobre a Amazônia no Youtube: Padrões de narrativa com o uso de Inteligência Artificial"*, a qual tem como autora a Cirino (2021), integrante do Grupo de Pesquisa Inovação e Convergência na Comunicação da Universidade Federal do Pará (UFPA).

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver uma ferramenta de extração de características de narrativas audiovisuais que permita estabelecer as relações existentes entre tais características com os principais elementos presentes em vídeos.

1.3.2 Objetivos específicos

A seguir serão apresentados os objetivos específicos para o desenvolvimento e conclusão desta proposta:

- Evidenciar características visuais para análise de narrativas audiovisuais, a partir da extração automática de tais características;
- Descrever o cenário/ambiente do vídeo a partir dos objetos presentes na cena;
- Observar o comportamento dos personagens do vídeo a partir das ações realizadas pelos mesmos;
- Verificar a existência de relação entre as características extraídas e os elementos da narrativa;
- Popular banco de dados com características de narrativas audiovisuais;
- Validar arquitetura com os processos de extração de características visuais.

1.4 Organização do Trabalho

Este trabalho tem a seguinte organização: no Capítulo 2 é apresentado a fundamentação teórica, onde se encontra os conceitos necessário para o entendimento da pesquisa, com destaque para descrição de cenas a partir da detecção, automática, de ações originadas das interações entre pessoas e objetos detectados em vídeos; já o Capítulo 3 possui a relação dos principais trabalhos utilizados como referência literária, para realização e conclusão da pesquisa; seguido pelo Capítulo 4, onde é descrito o problema que motivou este trabalho, assim como a proposta da solução para o mesmo; no Capítulo 5 é apresentada o processo metodológico utilizado para realização dos experimentos de validação da proposta, os quais são descritos no Capítulo 6, juntamente com as análises dos seus resultados; por fim, no Capítulo 7 são feitas as considerações finais da pesquisa, assim como as sugestões para trabalhos futuros.

2 Fundamentação Teórica

Este capítulo discorre sobre pontos fundamentais para o entendimento deste trabalho.

2.1 Detecção e Extração de Objetos de Frames

De acordo com Sourav Garg et al. (2020), de forma genérica, objetos são entidades ou coisas distintas presentes em um ambiente, os quais possuem a habilidade de serem vistos e tocados. Atualmente, as técnicas de *deep learning* têm sido muito usadas em aplicações que envolvem a tarefa de detecção de objetos, assim como em outras aplicações dentro da computação visual (RUSSEL; NORVIG et al., 2020a).

Em seu livro, Russel, Norvig et al. (2020b) afirmam que os modelos de detecção de objetos buscam diferentes objetos em uma dada imagem, reportam qual é a classe para cada objeto encontrado, e também retornam onde cada objeto se encontra no ambiente, através de um *bounding box* ao redor do objeto detectado. Por exemplo, Redmon e Farhadi (2018) propõem para o modelo detectar a classe dos objetos encontrados (i.e. cachorro, bicicleta, caminhonete, etc) e suas coordenadas de localização (*bounding-box*), conforme visto na Figura 1.

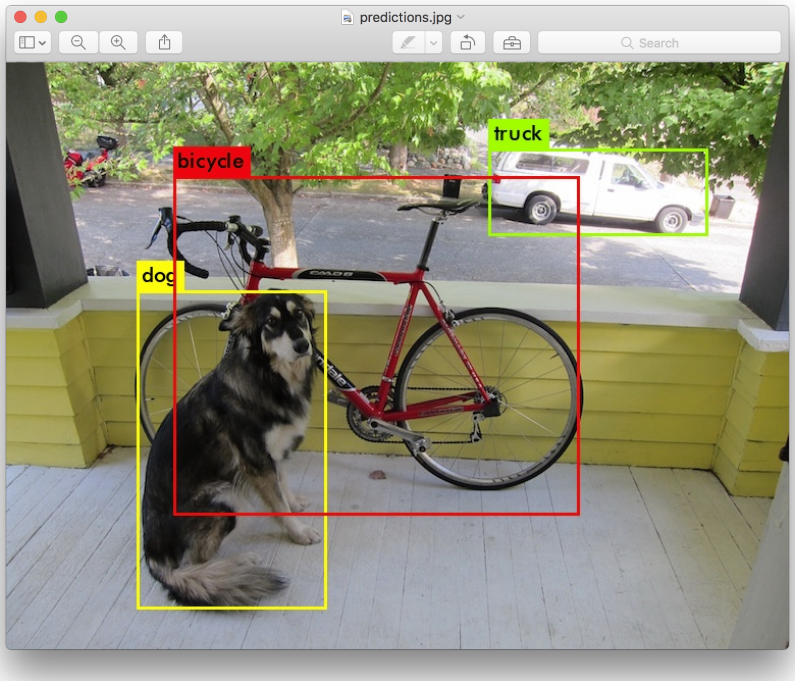


Figura 1 – Objetos detectados na imagem processada: 'dog' (*bounding boxe - amarelo*), 'bicycle' (*bounding boxe - vermelho*), 'truck' (*bounding boxe - verde*). Fonte: (REDMON; FARHADI, 2018).

2.1.1 YOLOv4: Detecção de Objetos em tempo real

Para realização do processo de detecção dos objetos, existem algumas soluções disponíveis no estado da arte. Uma delas é YOLOv4, que é uma versão aprimorada do modelo YOLO (*You Only Look Once* - Você só olha uma vez), desenvolvido por Joseph Redmon, et al. (2016). Ambas as versões possuem pesos pré-treinados, utilizando o *dataset dataset MS COCO*, podendo fazer predição de 80 classes diferentes (BOCHKOVSKIY; WANG; LIAO, 2020).

Resumidamente, o modelo faz uso de uma Rede Neural (Deep Convolutional Neural Network - Darknet) como extrator de características, que divide a imagem *input* em regiões menores e faz a predição dos *bounding boxes* para cada objeto detectado (REDMON; FARHADI, 2018), conforme pode ser visto na Figura 2.

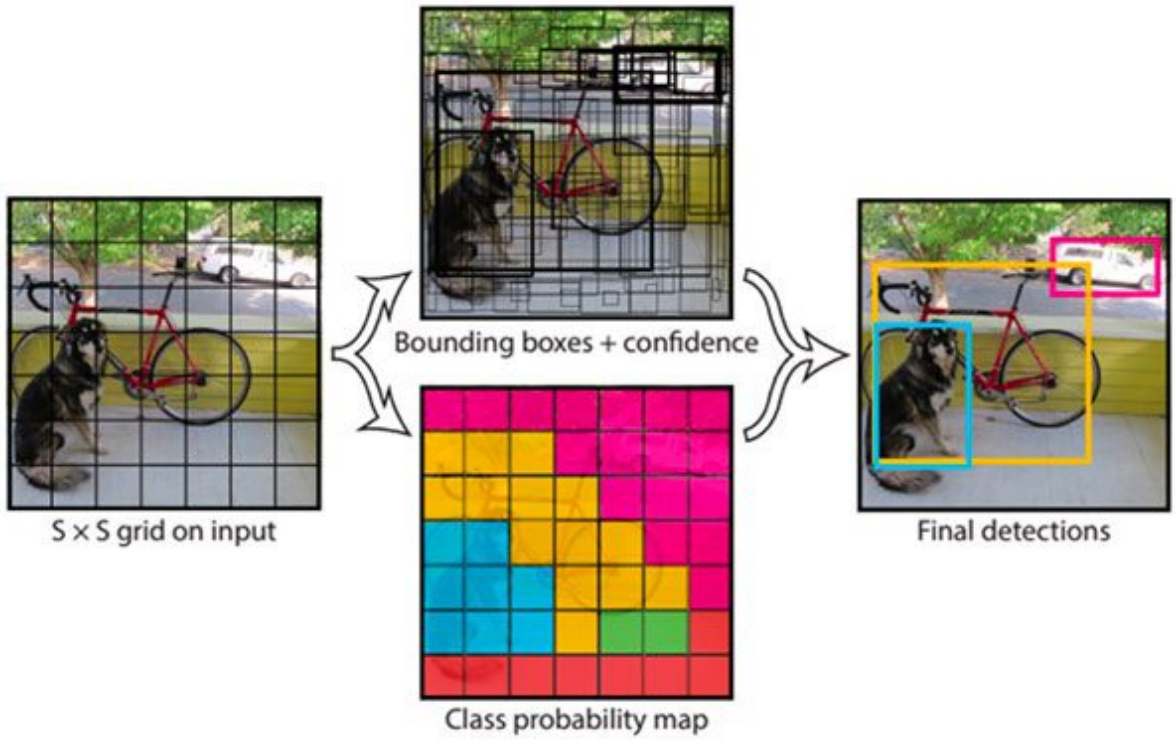


Figura 2 – Principais etapas presentes no processo de detecção de objetos pelo YOLO: Divisão da imagem em regiões menores (*S x S grid on input*), localização das características reconhecidas pelo modelo (*Bounding boxes + confidence*), imagem contendo os bounding boxes com maiores confianças (*final detections*). Fonte: (REDMON; FARHADI, 2018).

2.2 Detecção de ação em uma cena a partir da relação Pessoa-Objeto

Para compreensão e descrição de uma cena (sequência de *frames*) ao longo do intervalo de tempo de um vídeo, o telespectador (quem está assistindo), além do reconhecimento dos objetos de forma individual, também busca detectar como ocorrem as interações de pessoas com esses objetos.

De forma análoga, esse procedimento não é diferente para os modelos computacionais, visto que a representação de uma cena envolve a extração de informações visuais que podem sumarizar o conteúdo de uma imagem capturada, de uma cena específica (GKIOXARI et al., 2018), (GARG et al., 2020), pois, um vídeo é uma estrutura onde seus dados se apresentam de forma hierárquica, composta de eventos, cenas,

plano e *frames* (MAO et al., 2018).

Nesse contexto, a detecção das diferentes possibilidades de uso de um objeto em uma cena (*object's affordance*) pode ser tão importante quanto apenas identificar aquele objeto. O que possibilita obter sentido semântico *(e.g. 'write', 'drink')* em vez da categoria daquele objeto específico *(e.g. 'pen', 'cup')* (GARG et al., 2020).

A tarefa de detecção de interação entre pessoa e objeto (*Visual Relationship Detection*) tem o objetivo de detectar objetos (BOCHKOVSKIY; WANG; LIAO, 2020) e classificar tuplas (pessoa, predicado, objeto) em determinado imagem. Por exemplo, (pessoa, segura, caneta) seria um resultado dessa tarefa, o qual pode caracterizar o ato de escrever. Essa é uma tarefa importante para que um sistema inteligente consiga sumarizar e obter o sentido semântico de uma ação extraída de uma cena (CHIOU; ZIMMERMANN; FENG, 2021).

Portanto, modelos de reconhecimento de vídeo são responsáveis por permitir que um programa faça a classificação e detecção automática de atividades humanas que ocorreram ao longo do vídeo processado (GARG et al., 2020).

2.2.1 Detecção automática de ação em vídeos utilizando *PyTorchVideo*

Um vídeo é uma estrutura audiovisual, que traz consigo uma grande quantidade de dados, de diferentes modalidades (*i.e.* imagem, tempo, áudio). O que demanda que técnicas computacionais eficientes sejam utilizadas no processo de classificação e entendimento do seu conteúdo (FAN et al., 2021).

PyTorchVideo é uma biblioteca *open-source* de *deep learning*, que disponibiliza componentes e recursos para as variadas aplicações de automatização da classificação de vídeos, incluindo a detecção de atividades/ações reproduzidas por pessoas. Essa biblioteca possui diferentes *frameworks*, baseados em Python, que facilitam o seu uso na aplicação de modelos pré-treinados, como o *SlowFast* (FEICHTENHOFER et al., 2019), utilizado nesta proposta. A biblioteca com a sua documentação está disponível no *GitHub* em <https://github.com/facebookresearch/pytorchvideo> (FAN et al.,

2021).

2.3 Narrativas audiovisuais e seus elementos

As narrativas audiovisuais são construídas a partir da combinação de som, imagem e palavras. Esses elementos, com a interferência de outros, criam mensagens para transmitir informações, opiniões, ideias, sensações e sentimentos que são percebidos pelos seus espectadores, na criação de sentidos e significados (GUIMARÃES, 2022). Segundo Jacques Aumont (1995), a imagem existe para ser vista por um espectador historicamente definido. Ela é sempre produzida por um agente com o objetivo de dirigir-se a um outro, com uma mensagem determinada e condicionada por fatores sociais, culturais, técnicos e ideológicos.

Os recursos narrativos são aqueles elementos centrais da narrativa, necessários para que ela exista, ou seja, para que ela seja contada por alguém, que conta acontecimentos e só os pode contar porque estes já terminaram. Além disso, um grupo de elementos significativos têm, também, o papel de construir a narrativa audiovisual, onde, seus elementos, quando arrumados de forma estratégica, produzem determinados efeitos de sentido (CIRINO, 2021).

Em seu trabalho de pré-qualificação, a pesquisadora Cirino (2021) também afirmou que os recursos visuais são constituídos basicamente pelas imagens compostas pelos objetos e pessoas presentes em um vídeo. Os recursos sonoros envolvem todo o tipo de som, naturais e artificiais, provocando no espectador uma experiência sensorial. O movimento, proporcionado tanto através da imagem como através do som, é o elemento fundamental da narrativa audiovisual, situando quem o consome no tempo e no espaço, a partir do ritmo e do que é visível (CIRINO, 2021).

Conforme descrito pela Guimarães (2022), a narrativa é composta por 6 principais campos: foco narrativo, ação, personagem/pessoa, linguagem, tempo, e espaço/ambiente. Nesses campos há diferentes elementos que são usados para caracterização da narrativa audiovisual. Características essas que podem ser vistas na Tabela 1, a qual foi montada com base no conceito de narrativa definido por Luiz Gonzaga Motta (2005).

Tabela 1 – Elementos para caracterização e composição da narrativa: FOCO NARRATIVO, AÇÃO, PERSONAGEM/PESSOA, LINGUAGEM, TEMPO, ESPAÇO/AMBIENTE. Fonte: (GUIMARÃES, 2022).

FOCO NARRATIVO				**AÇÃO**	**PERSONAGEM/PESSOA E AÇÃO**		
				Como é dito?	**Quem diz?**		
Tipo de narrativa	Temática	Contexto ou dimensão	Metadados	Interação objeto dos personagens	Expressões faciais	Gestos e movimentos	Entonação e intensidade de voz
LINGUAGEM			**TEMPO**			**ESPAÇO/AMBIENTE**	
O que e como é dito?			**Quando é dito?**			**Onde é dito?**	
Classificação gramatical das palavras	Classificação de polaridade	Discurso	Duração do vídeo	Duração análise da narrativa	Data de postagem do vídeo	Cena	Cenário	Objetos

3 Trabalhos Relacionados

A seguir, na Tabela 2, estão listados os trabalhos, encontrados na literatura, cuja principal fonte de análise é o vídeo. Esses são os principais trabalhos relacionados com esta proposta.

Tabela 2 – Quadro comparativo dos principais trabalhos relacionados com a pesquisa. Fonte: Própria (2022).

	Trabalho	Extração de frames e metadados de vídeos do YouTube	Detecção e Extração de Objetos de Frames	Detecção automática de ação em vídeos	Elementos de Narrativas Audiovisuais
1	A Amazônia e Polarização Política no Youtube: Representação de Narrativas com o Uso de Inteligência Artificial (CIRINO et al., 2021).	SIM	NÃO	NÃO	SIM
2	Identificação de características de Aspectos Emocionais Associados a Elementos de Narrativas Audiovisuais (GUIMARÃES, 2022).	SIM	NÃO	NÃO	SIM
3	YOLOv4: Optimal Speed and Accuracy of Object Detection (BOCHKOVSKIY; WANG; LIAO, 2020)	NÃO	SIM	NÃO	NÃO
4	Detectron2: Facebook AI Research's software system that implements state-of-the-art object detection (WU et al., 2019)	NÃO	SIM	NÃO	NÃO
5	PyTorchVideo: A Deep Learning Library for Video Understanding (FAN et al., 2021)	NÃO	NÃO	SIM	NÃO
6	SlowFast Networks for Video Recognition (FEICHTENHOFER et al., 2019)	NÃO	SIM	SIM	NÃO
7	Semantics for Robotic Mapping, Perception and Interaction: A Survey (GARG et al., 2020)	NÃO	SIM	SIM	NÃO

No primeiro trabalho, listado acima, encontra-se o artigo da Cirino et al. (2021), onde a pesquisa tinha como principal objetivo a identificação e análise de padrões de narrativas presentes em vídeos publicados em canais políticos do YouTube. Para isso, um processo, utilizando técnicas computacionais e a API do YouTube, foi desenvolvido, para realizar a extração de dados textuais e metadados dos vídeos usados para análise.

Em comparação a este trabalho, além da utilização da mesma API para extração de dados de vídeos, publicados na mesma plataforma, também é realizado a extração automática de características visuais, através da utilização de modelos pré-treinados de inteligência artificial. Características essas que estão relacionadas com os principais elementos que caracterizam uma narrativa audiovisual.

Já no segundo trabalho da lista, cujo título é "Identificação de Características de Aspectos Emocionais Associados a Narrativas Audiovisuais", Guimarães (2022) propôs a identificação de características visuais com aspectos emocionais, também associadas aos principais elementos de narrativas audiovisuais. Porém, essa proposta se limita apenas à detecção de características das pessoas detectadas nos vídeos, uma vez que o estudo das características emocionas, naquele caso, é feito através do reconhecimento

de expressões faciais.

Enquanto que, neste trabalho, mesmo que fazendo a detecção da presença de pessoas em cenas dos vídeos, é feito, também, a identificação e a extração de características que se relacionam com elementos que caracterizam o ambiente, o cenário e o movimento/ação de uma narrativa audiovisual, conforme foi descrito pela autora, mencionada acima.

Com relação aos trabalhos de Bochkovskiy, Wang e Liao (2020) (3) e Wu et al. (2019) (4), os pesquisadores apresentam o desenvolvimento de modelos de detecção de objetos em imagens. Detectron2 é uma biblioteca, desenvolvida pelo grupo de pesquisas Facebook, que disponibiliza algoritmos, no estado da arte, para detecção e segmentação de objetos. Yolov4 também é uma solução disponível no estado da arte para detecção de objetos. Porém, nenhum deles abordam ou propõem a utilização dos objetos detectados no processo de caracterização de narrativas audiovisuais, como é proposto neste trabalho. Para a solução proposta neste trabalho, optou-se por utilizar o modelo pré-treinado disponibilizado pelo YoloV4, mas os modelos disponibilizados pelo trabalho Detectron2 também poderiam ser utilizados no processo de detecção de objetivos, uma vez que esse tem o mesmo objetivo proposto pelos pesquisadores Bochkovskiy, Wang e Liao (2020).

Quanto à tarefa de detecção automática, de acordo com o relacionado na Tabela 2, temos os trabalhos de Fan et al. (2021) e Feichtenhofer et al. (2019), onde ambos apresentam redes neurais para realizar tarefas de reconhecimento de vídeos. De acordo com o descrito, os modelos disponibilizados atingem uma performance considerável para classificação e detecção de ação. PyTorchVideo é uma biblioteca open-source de deep learning, que disponibiliza componentes e recursos para as variadas aplicações de automatização da classificação de vídeos, incluindo a detecção de atividades/ações reproduzidas por pessoa, dentre eles têm os modelos slowfast. Com relação a este trabalho, além da utilização do modelo pré-treinado, disponibilizado pelo slowfast através da biblioteca Pytorchvideo, também é proposto a utilização das ações detectadas no vídeos no processo de caracterização de narrativas audiovisuais.

4 Problema e Proposta da Solução

4.1 Problema

O esforço humano gasto no processo de análise e caracterização de narrativas audiovisuais é inimaginável dada a enorme quantidade de vídeos compartilhados diariamente na internet. Principalmente pelo fato do vídeo ser uma estrutura de dados multimodal, dificultando a extração de características de forma manual, pois isso demandaria trabalho de um grupo muito grande de especialistas executando somente a tarefa de extração de tais características, para então iniciar a fase de caracterização da narrativa audiovisual. Portanto, a automatização deste processo torna-se imprescindível para lidar com os problemas que o conteúdo, enviesado ou mesmo falso, de tais vídeos podem trazer.

4.2 Proposta

Este trabalho tem como proposta o desenvolvimento de uma ferramenta, desenvolvida em Python, para extração automática dos elementos visuais que caracterizam o ambiente e as ações/interações, necessários para caracterização e análise de narrativa audiovisuais. Para isso, a utilização de modelos pré-treinados de detecção de objetos e detecção de ação/atividades será incluído no processo, uma vez que esses modelos extraem automaticamente essas características, que compõem os recursos visuais de um vídeo. O que torna possível a integração desta ferramenta no processo de análise de narrativas audiovisuais, otimizando, assim, o tempo de realização de extração dos dados dos vídeos (Seção 2).

A Figura 3 ilustra os principais processos presentes na arquitetura da ferramenta proposta, assim como o fluxo dos dados resultantes de cada etapa. Nesse cenário, optou-se por utilizar a plataforma YouTube como fonte de coleta dos dados, uma vez que, de acordo com Carta et al. (2022), é a segunda plataforma de compartilhamento de vídeo mais utilizada atualmente.

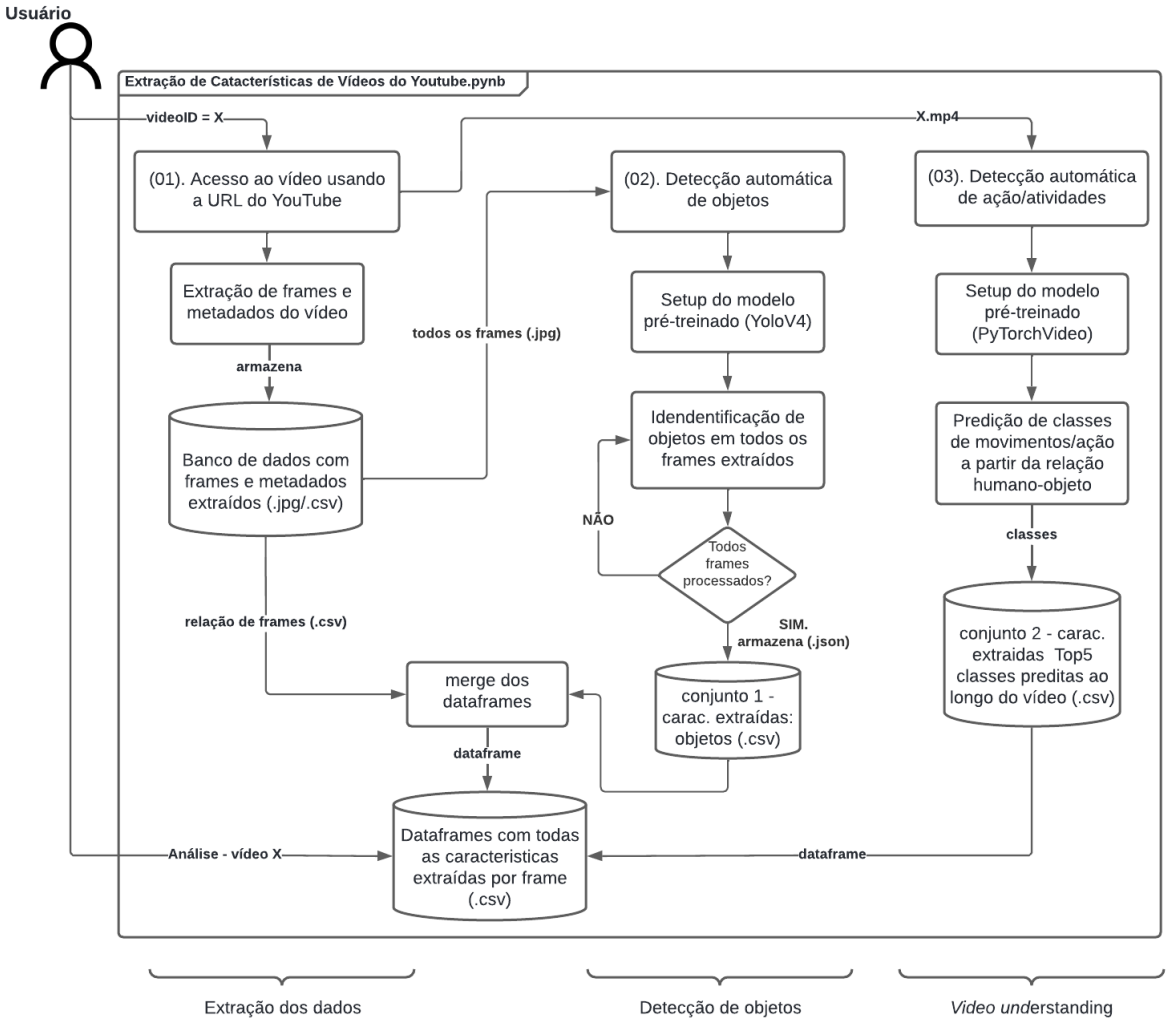


Figura 3 – Arquitetura com os principais processos executados pela ferramenta proposta para extração de características visuais de vídeo do YouTube. São eles: (01) Extração de dados, (02) detecção de objetos, (03) detecção de ações. Fonte: Própria (2022).

Seguindo os processos ilustrados na Figura 3, após o usuário definir o vídeo amostra (videoID=X), a ferramenta, inicialmente, acessa, automaticamente, o vídeo a ser processado e faz a extração dos *frames* e metadados, os quais são, posteriormente, armazenados em um banco de dados. E um primeiro *dataframe*, contendo os nomes e o tempo de ocorrência de cada *frame*, é criado e armazenado.

Em seguida, a ferramenta inicia o processo de configuração do modelo, pré-treinado, que será utilizado para detecção automática de objetos nos *frames* extraídos anteriormente. Para isso, é executada uma função que acessa os *frames*, armazenados no

banco de dados, a qual retorna uma lista com os caminhos de armazenamento de cada arquivo .jpg. Os quais são processados, um a um, pela função de predição de objetos pelo modelo usado. No final desse processo, um arquivo .json, que contém a informação de processamento de cada imagem, é acessado e manipulado para ser feita a extração dos objetos relevantes, juntamente com a frequência de ocorrência de cada um.

Após as duas etapas anteriores, com o objetivo de centralizar as características extraídas por *frame* processado, um novo *dataframe* é criado. Esse será resultante do *merge* do primeiro *dataframe*, que contém o nome e o tempo que cada *frame* ocorreu no vídeo, com um dicionário que contém os objetos detectados como as chaves dessa estrutura de dados, junto de seu número de ocorrências, como valor.

A última fase de processamento da ferramenta é a chamada *video understanding*. Onde, após o *setup* do modelo *slow_r50_detection*, disponibilizado pela PytorchVideo, é iniciado o processo de detecção automática de ação/atividades no arquivo .mp4, o qual foi obtido através do objeto de acesso da *url* do YouTube, criado na primeira fase da arquitetura proposta. Como resultado desse processamento, espera-se obter as *top* 3 classes correspondentes as atividades detectadas na cena do vídeo processado. Essas classes das ações detectadas são escritas em um arquivo .csv e, então, armazenado. Com esse resultado, é esperado que o usuário da ferramenta consiga classificar e/ou descrever o vídeo de entrada (videoID=X), a partir das atividades predominantes realizadas ao longo do mesmo.

Todas as características extraídas são escritas em *dataframes* específicos do vídeo processado. E, então, salvos como arquivos .csv no banco de dados. Os quais estarão disponível para manipulação tanto pelo usuário, quanto para futura pesquisas de aprimoramento da ferramenta proposta ou para estudos que estejam relacionados com a classificação de vídeos.

5 Metodologia

Nesta seção é apresentada a metodologia adotada para realização do desenvolvimento desta pesquisa. Ela está dividida da seguinte forma:

5.1 Extração e armazenamento de dados de vídeos do Youtube

Como ponto de partida para execução da ferramenta proposta, é necessário a definição do vídeo que será processado pela mesma. Como já mencionado anteriormente, os vídeos utilizados, necessariamente, precisam estar publicados na plataforma do Youtube. Pois, a ferramenta espera, como entrada, o valor do identificador único do vídeo (videoId), além do intervalo desejado para análise (tempoInicial e tempoFinal), conforme visto na Figura 4.

Após a definição do vídeo, para extração das características propostas, é dado início à primeira parte desta metodologia: extração e armazenamento dos dados do vídeo de entrada. Para esta tarefa, os processos definidos no diagrama da Figura 5 são seguidos para realização da extração dos dados, que serão utilizados como entradas das próximas etapas da metodologia. Nesta etapa, três principais bibliotecas públicas, implementadas em Python, são instaladas. São elas: Pafy, Pandas e OpenCV.

```
[14]  1 # passar o id do vídeo
      2 videoId = 'Mh4f9AYRCZY'
      3 #intervalo do vídeo para análise:
      4 tempoInicial = '0:00:00'
      5 tempoFinal = '0:01:00'
      6 #pasta onde os frames e caracteristicas extraidas serão salvos
      7 folderPath = "/content/video0"
```

```
[22]  1 # This is the video we're going to process
      2 print("This is the video we're going to process")
      3 from IPython.display import YouTubeVideo, display
      4 video = YouTubeVideo(videoId, width=500)
      5 display(video)
```

```
This is the video we're going to process
```

Figura 4 – Resultado da execução do script de verificação e definição do vídeo e do intervalo de tempo que será processado pela ferramenta proposta. Fonte: Própria (2022).

A biblioteca Pafy permite o download do conteúdo do vídeo, diretamente do YouTube, através de sua *url*, além da recuperação dos metadados daquele vídeo (PAFY, 2019). A biblioteca Pandas (REBACK et al., 2022) é utilizada para realizar a manipulação dos dados recuperados da API do Youtube (YOUTUBE, 2022), para, então, gerar e relacionar os *dataframes*. Por fim, a OpenCV foi utilizada para manipulação dos arquivos de imagens extraídas e salvas no banco de dados (OPENCV, 2015).

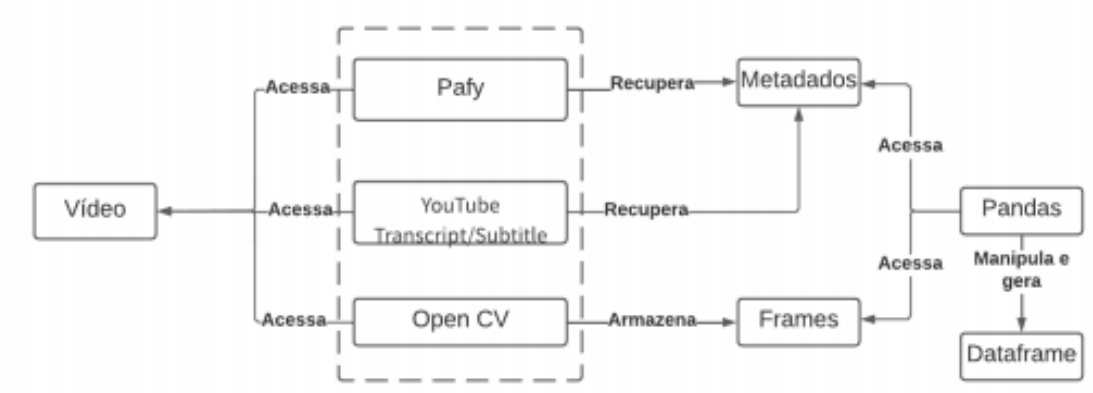


Figura 5 – Diagrama com o funcionamento do processo de extração de frames e metadados de videos do Youtube. Fonte: (GUIMARÃES, 2022).

O *frame* é um quadro do vídeo que contém uma imagem fixa. O conjunto de vários *frames* constitui uma cena do vídeo. De acordo com Victória Guimarães (2022), a retirada de 1 (um) *frame* por segundo (FPS) pode otimizar o processo de execução dos *scripts* de processamento de imagens, além de não impactar na análise final do conteúdo visual, uma vez que, no intervalo de 1 segundo, os *frames* são similares.

Levando isso em consideração, é feita a extração de 1 FPS do vídeo *input*, e, posteriormente, esse conjunto de *frames* originais são armazenados no banco dados. Além disso, com o auxílio da biblioteca Pandas, as informações com o nome do *frame* extraído e o tempo em que esse mesmo ocorre, são relacionados e escritos em arquivo .csv. Conforme ilustrado na Figura 6.

Figura 6 – *Pipeline* dos principais processos para extração e armazenamento dos dados do vídeo *input*. São eles: acesso ao vídeo publicado no YouTube, extração e armazenamento dos frames e metadados presentes no vídeo. Fonte: Própria (2022).

5.2 Detecção de objetos nos frames extraídos do vídeo

Trata-se do processo de detecção e extração de objetos presentes em uma determinada imagem (*frame*). Para isso, foi desenvolvido um código, em python, para executar todo o processo, desde o acesso aos frames até à organização dos objetos detectados em um dataframe. Como já mencionado, para a tarefa de detecção de objeto optou-se por utilizar o YoloV4, para isso, foi seguido o tutorial disponível em

<https://github.com/AlexeyAB/darknet>.

Vale ressaltar que neste trabalho não foi feita a avaliação de desempenho do modelo utilizado. Uma vez que não é foco desta pesquisa fazer nenhum tipo de treinamento de modelos de detecção de objetos. Apenas foram utilizados os arquivos de configuração (*i.e.* /cfg/yolov4.cfg e yolov4.weights) de uma arquitetura pública, cujo treinamento e validação/avaliação da mesma foi feita por outros pesquisadores, conforme pode ser visto em <https://github.com/AlexeyAB/darknet#pre-trained-models>. Neste caso, o modelo utilizado apresentou uma média de AP (média de precisão) de 65,7% (mAP) [1].

Dito isto, após realizado a execução da etapa de configuração do modelo pré-treinado, o que, basicamente, é feita executando os blocos de códigos referentes à isso, a tarefa de detecção se inicia com o acesso ao diretório de *frames*, extraídos nas etapas anteriores; seleciona a primeira imagem (imgX.jpg) e a utiliza como *input* para o processo de predição das classes, pelo YoloV4. Como pode ser visto na Figura 7, essa fase do processo retorna como resultado (*output*) as classes dos objetos detectados com sua respectiva confiança de classificação. As classes detectadas, acompanhadas de sua frequência de ocorrência, são escritas em um dicionário, que, posteriormente, é adicionado no *dataframe* (Figura 17) para ser armazenada no banco de dados.

[1] *mAP (mean average precision)*: é a média do valor de AP (Average precision) de todas as clases. Onde AP é a média de precisão (TP/TP+FP), ou seja, é a média de predições corretas, feitas pelo modelo treinado. Fonte:(YOHANANDAN, 2020)

Figura 7 – Classes de objetos preditos pelo modelo YoloV4: {'book': 6, 'person': 2, 'chair': 1, 'tie': 1}. Fonte: Própria (2022).

Finalizado toda a fase de processamento do primeiro *frame*, o processo se repete para o próximo *frame* (imgX+1.jpg) e segue assim até o processamento de todos os *frames* presentes no banco de dados.

index	tempo	frame	frames_path	objetos
0	0:00:00	img1.jpg	/content/video0/img1.jpg	
1	0:00:01	img2.jpg	/content/video0/img2.jpg	{'book': 17, 'chair': 1, 'bottle': 1, 'tie': 1, 'person': 1}
2	0:00:02	img3.jpg	/content/video0/img3.jpg	{'book': 21, 'person': 2, 'chair': 1, 'bottle': 1, 'tie': 1}
3	0:00:03	img4.jpg	/content/video0/img4.jpg	{'book': 17, 'person': 2, 'chair': 1, 'bottle': 1, 'tie': 1}
4	0:00:04	img5.jpg	/content/video0/img5.jpg	{'book': 11, 'person': 2, 'chair': 1, 'bottle': 1, 'tie': 1}
5	0:00:05	img6.jpg	/content/video0/img6.jpg	{'book': 11, 'person': 2, 'chair': 1, 'bottle': 1, 'tie': 1}
6	0:00:06	img7.jpg	/content/video0/img7.jpg	{'person': 2, 'tvmonitor': 1, 'cup': 1, 'bottle': 1, 'tie': 1}
7	0:00:07	img8.jpg	/content/video0/img8.jpg	{'person': 3, 'tvmonitor': 1, 'chair': 1, 'bottle': 1, 'tie': 1}
8	0:00:08	img9.jpg	/content/video0/img9.jpg	{'person': 3, 'tvmonitor': 1, 'chair': 1, 'cup': 1, 'bottle': 1, 'tie': 1}

Figura 8 – *Dataframe* com as classes de objetos detectados para cada *frame* do vídeo. Este é usado para relacionar o tempo em que um frame foi extraído e armazenado (frame_path) junto com a sua relação de objetos detectados. Fonte: Própria (2022).

5.3 Detecção automática de ação/atividade em vídeo

Este processo consiste na execução do código, também desenvolvido em *Python*, que realiza a tarefa de detecção de atividades em vídeos. Para isso é utilizado o modelo,

pré-treinado, *slow_r50_detection*, disponibilizado pela biblioteca *PyTorchVideo*. A qual, combinada com outras bibliotecas públicas, facilita o processo de extração dessas características, que podem ser utilizadas como elementos para descrição de cenas presentes naquele vídeo. (PYTORCHVIDEO, 2021).

Conforme já mencionado, não foi executado nenhum processo de avaliação de desempenho do modelo utilizado, apenas foi utilizado os arquivos de configuração do modelo executado neste processo, o qual, de acordo com o disponível em <https://github.com/facebookresearch/pytorchvideo/blob/main/docs/source/model_zoo.md>, tem uma acurácia de 72,4%.

Sendo assim, inicialmente é realizado a execução das etapas de *setup* inicial do ambiente de execução, as quais são necessárias para que a fase de predição, realizada pelo modelo utilizado, ocorra corretamente, para isso foi seguido o tutorial definido em <https://pytorchvideo.org/docs/tutorial_torchhub_detection_inference>. Em seguida, utilizando, como entrada da fase de predição, o vídeo armazenado, o qual foi baixado diretamente do vídeo do YouTube, em formato .mp4, utilizando a *Pafy*, é iniciado o processamento do vídeo.

Onde, primeiro é gerado o *bounding box* para cada pessoa detectada no *frame* (utilizando a função *get_person_bboxes()*) e, então, é inicia a predição das ações/atividades para cada pessoa detectada, na imagem processada. A Figura 9 é um exemplo de parte do resultado deste processamento, onde a mesma contém os *bounding boxes*, na cor verde, para cada pessoa; acompanhada das *labels* detectadas por bounding boxes, as quais representam as ações detectadas para aquele momento.

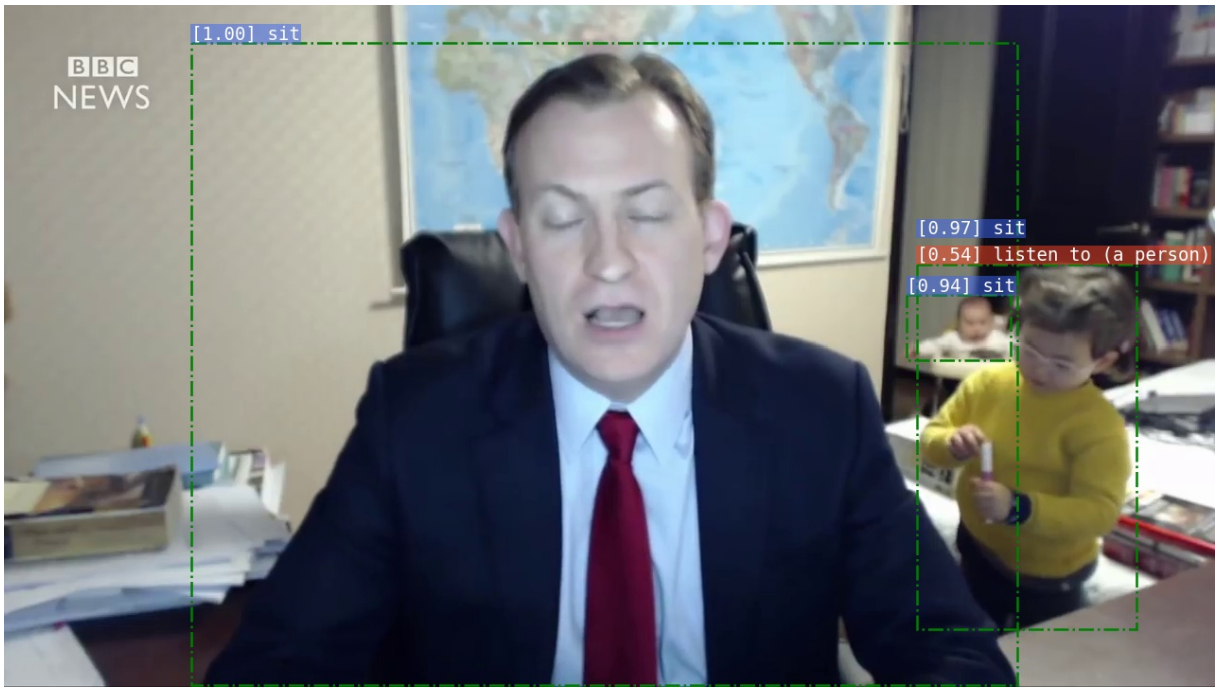


Figura 9 – Atividades detectadas para cada pessoa localizada no frame processado: {pessoa1: ["[1,00] *sit*"], pessoa2: ["[0.97] *sit*", "[0.54] *listen to (a person)*"], pessoa3: ["[0.94] *sit*"]}. Fonte: Própria (2022).

Seguido do processamento de todo o intervalo de tempo definido para o vídeo utilizados para predição, esta fase retorna como resultado final um vídeo *output*, montado com todos os *frames* processado, além de um arquivo .csv contento as classes das atividades detectadas por *frame*. Isso é salvo no banco de dados, para que seja utilizado para entendimento e descrição conteúdo do vídeo amostra.

5.4 Elementos da narrativa audiovisual relacionados com as características extraídas do vídeo

De acordo com a Tabela 1, o espaço/ambiente de uma narrativa pode ser caracterizado pela **cena**, **cenário** e **objetos** presentes em um vídeo. Além disso, foi possível observar que a **interação personagem e objeto** caracterizam uma **ação** que compõe uma narrativa.

Desta forma, levando em consideração o que foi definido na Tabela 1, está pes-

quisa se propõem a automatizar o processo de identificação e extração das características que compõem uma narrativa a partir da descrição dos campos ação e espaço/ambiente. Ambos se utilizam dos objetos presentes nos vídeos e das interações entre pessoa e objeto, para caracterizar uma narrativa audiovisual (MOTTA, 2005).

Assim, a relação entre as características extraídas, nas fases anteriores, do vídeo, e as características que compõem uma narrativa audiovisual pode ser vista na Tabela 3.

Tabela 3 – Relação das características extraídas utilizado modelos pré-treinados presentes em vídeos com os elementos da narrativa audiovisual. Fonte: Própria (2022).

	ESPAÇO/AMBIENTE	**PERSONAGEM/PESSOA E AÇÃO**
Características para a composição da narrativa	**Objetos**, Cena e Cenário	Interação objeto x pessoa
Método de extração automática	Modelo pré-treinado - YOLOv4	Modelo pré-treinado - PyTorchVideo

6 Experimentos e Resultados

Para os experimentos de validação dos processos presentes na arquitetura da ferramenta proposta, foram escolhidos 4 vídeos, publicados no YouTube, para ser usados como vídeo entrada da ferramenta desenvolvida. Com o objetivo de limitar o escopo da pesquisa, todos esses vídeos fazem parte de uma mesma categoria de vídeos da plataforma, a categoria *"News & Politics"*.

Além disso, para centralizar os esforços na execução e validação do código desenvolvido e, também, para facilitar o processo de configuração das dependências necessárias para utilização do mesmo, optou-se por utilizar as máquinas virtuais e o ambiente de execução *online*, disponibilizado pela plataforma Google Collaboratory, um ambiente de desenvolvimento Python que é executado diretamente no navegador comum (Google Chrome, Firefox, etc) (BISONG, 2019). Os dados extraídos foram armazenados no Google Drive.

Considerando os fatos mencionados acima, a seguir serão descritos os experimentos realizados, juntamente com os resultados obtidos durante a realização dos mesmos.

6.1 Vídeos para realização dos experimentos

A lista dos vídeos (Tabela 4) que serão utilizadas para validação da propostas, através dos experimentos, foi montada a partir das considerações e pesquisas realizadas pelo Grupo de Pesquisa de Inteligência Artificial da UFAM em parceria com o Grupo de Pesquisa Inovação e Convergência na Comunicação da Universidade Federal do Pará (UFPA). Esses vídeos foram retirados do conjunto de vídeos levantados e analisados por Cirino et al. (2021), no artigo *A Amazônia e Polarização Política no **YouTube**: Representação de **Narrativas** com o uso de Inteligência Artificial*.

Tabela 4 – Vídeos do YouTube usados para extração das características propostas. Fonte: Própria (2022).

	VideoID	Título do Vídeo (duração: mm:ss)	Intervalo de interesse	Duração Total do intervalo (s)
1	prkZ-s8jP5g	'Live da semana com Presidente Jair Bolsonaro - 22/04/2021. Temas na descrição' (42:14)	12'43"a 14'10"	87
2	qEd8Y0pi5_4	'AoVivo: Abertura da 76ł Sessão da Assembleia-Geral da ONU' (14:37)	01'15"a 2'20"	65
3	2WesQczDivs	'Chefe da CIA visita Bolsonaro em encontro reservado e Presidente diz que pode haver um vale-tudo' (26:16)	de 10"a 01'18"	68
4	2WesQczDivs	'Jovem indígena testa positivo para Covid-19' (02:51)	de 8"a 02'51"	242

6.2 Extração de Característica dos vídeos

Nesta seção é apresentado e descrito os principais resultados obtidos da execução dos processos de extração das características visuais dos vídeos listados na Tabela 4. A execução dos processos foi realizada de forma individual para cada um dos 4 vídeos, seguindo o que foi descrito na parte metodológica deste trabalho, conforme pode ser visto nas Figuras 10, 11, 12, 13.

Figura 10 – Resultado da execução do script de verificação e definição do vídeo e do intervalo de tempo retornados para o vídeo 1. Fonte: Própria (2022).

Figura 11 – Resultado da execução do script de verificação e definição do vídeo e do intervalo de tempo retornados para o vídeo 2. Fonte: Própria (2022).

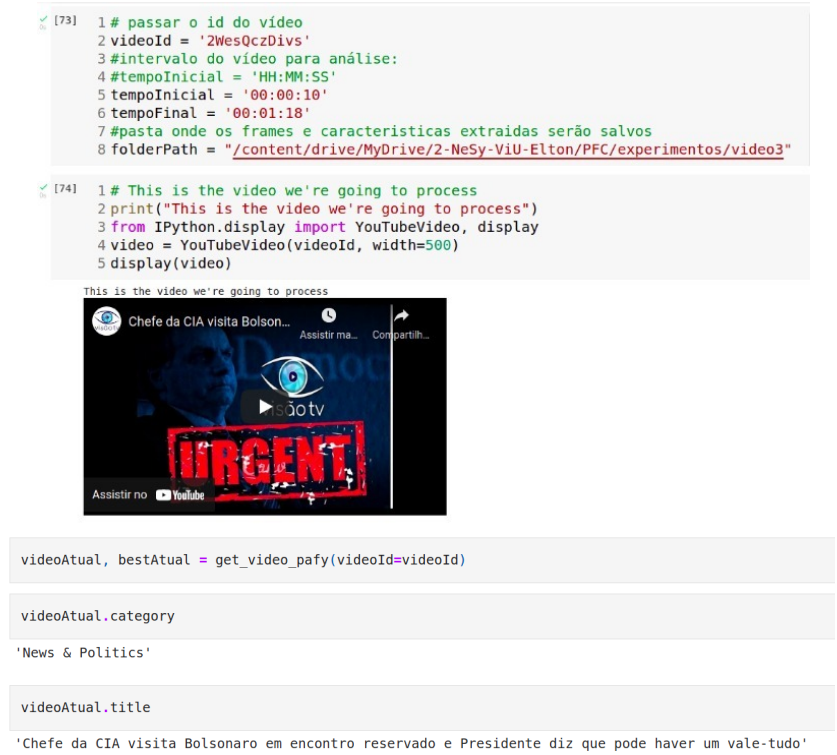


Figura 12 – Resultado da execução do script de verificação e definição do vídeo e do intervalo de tempo retornados para o vídeo 3. Fonte: Própria (2022).

Figura 13 – Resultado da execução do script de verificação e definição do vídeo e do intervalo de tempo retornados para o vídeo 4. Fonte: Própria (2022).

Após definidos os identificadores únicos (videoId) e o intervalo de tempos de análise (tempoInicial e tempoFinal), foi possível validar essa etapa inicial de acesso ao vídeo. Onde foi retornado a tela inicial de cada vídeo e os atributos de publicação (categoria e título) para os mesmos. Conforme pôde ser visto nas figuras menciondas acima.

6.2.1 Extração de *frames* dos vídeos

Conforme ilustrado na Figura 14, o processo de extração de *frames* funcionou conforme o esperado. Para o quarto vídeo da Tabela 4, 164 *frames* foram extraídos, uma vez que esse foi o vídeo com maior intervalo de análise, mesmo sendo o vídeo com menor duração.

Figura 14 – Resultado contendo a quantidade de *frames* extraídos por vídeo. [Vídeo 01: 87 frames], [Vídeo 02: 65 frames], [vídeo 03: 68 frames], [vídeo 04: 164 frames]. Fonte: Própria (2022).

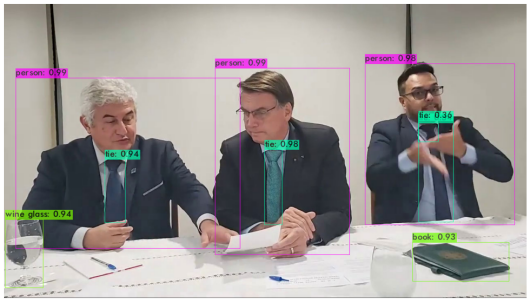
Além disso, ainda durante o processo de extração dos *frames* do vídeo 1, percebeu-se que quanto maior o valor definido como tempo inicial (tempoInicial), para leitura dos frames do vídeo, maior será o tempo de espera de finalização desse processo. Por exemplo, levou cerca de 495s (*i.e.* 8,25 minutos) para processar e armazenar todos os *frames* desejados (Figura 15 (a)); isso porque a função passou por 763 * 29 *frames*, para começar a salvar os *frames* lidos a partir do tempo inicial 12:43. Enquanto que, para o vídeo 4, onde o tempo inicial era igual à 08 segundos, o processo levou somente 25,2s (Figura 15 (b)) para finalizar o processo de leitura e armazenamento de 164 *frames*.

(a) Tempo de execução no vídeo 1. (b) Tempo de execução no vídeo 4.

Figura 15 – Tempo de execução (s) do processo de extração dos *frames* do vídeo 1 (com tempo de duração = 42’12”) e dos *frames* do vídeo 4 (tempo de duração = 2’51”). Fonte: Própria (2022).

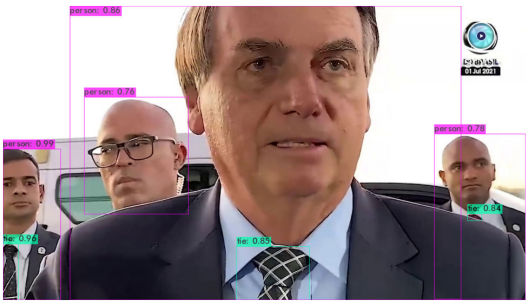
6.2.2 Detecção automática de objetos presentes nos *frames*

Através do endereço de armazenamento dos *frames*, extraídos de cada vídeos, o processo de detecção de objetos foi executado para cada uma dessas imagens. Na Figura 16, pode-se ver o resultado da detecção de objetos presentes em uma dos *frames* de cada vídeo processado. Isso indica que, caso haja necessidade, a ferramenta proposta pode, também, retorna os *frames*, pós-processados, com os objetos detectados devidamente localizados, através de seus *bounding-boxes*.



(a) Objetos detectados no vídeo 1

(b) Objetos detectados no vídeo 2



(c) Objetos detectados no vídeo 3

(d) Objetos detectados no vídeo 4

Figura 16 – Ilustração dos objetos detectados em um dos *frames* dos 4 vídeos. Fonte: Própria (2022).

Quanto a análise das classes de objetos detectadas, por imagem processada, foi observado, através da manipulação do *dataframe* resultante dos *frames* do vídeo 1, que há muita repetição de resultado, uma vez que esse é um vídeo com cenário fixo e a captação é feita por uma única câmera, isso pode ser confirmado na Figura 17, onde o resultado *{'tie': 3, 'person': 3, 'book': 1, 'wine glass': 1}* aconteceu para 36 frames.

Ainda em relação aos objetos detectados no vídeo 1, foi confirmado que houveram objetos que não foram detectados (*e.g.* 'caneta', 'papel', 'óculos') (Figura 16 (a)), pois essas classes não fazem parte do *dataset* utilizado como conjunto de treinamento

do modelo de predição executado.

objetos detectados	**Nº de frames**
{'tie': 3, 'person': 3, 'book': 1, 'wine glass': 1}	36
{'tie': 4, 'person': 3, 'book': 1, 'wine glass': 1}	14
{'tie': 3, 'person': 3, 'book': 1, 'cup': 1, 'wine glass': 1}	12
{'tie': 4, 'person': 3, 'book': 1, 'cup': 1, 'wine glass': 1}	4
{'tie': 3, 'person': 3, 'book': 1, 'diningtable': 1, 'wine glass': 1}	3
{'person': 3, 'tie': 2, 'book': 1, 'wine glass': 1}	3
{'tie': 3, 'person': 3, 'book': 2, 'wine glass': 1}	2
{'tie': 3, 'person': 3, 'book': 1, 'chair': 1, 'cup': 1, 'wine glass': 1}	2
{'tie': 6, 'person': 3, 'book': 1, 'wine glass': 1}	1
{'tie': 5, 'person': 3, 'book': 1, 'wine glass': 1}	1
{'tie': 4, 'person': 3, 'book': 2, 'cup': 1, 'wine glass': 1}	1
{'tie': 3, 'person': 3, 'book': 2, 'cup': 1, 'wine glass': 1}	1
{'tie': 3, 'person': 3, 'book': 1, 'laptop': 1, 'cup': 1, 'wine glass': 1}	1
{'tie': 3, 'person': 3, 'book': 1, 'diningtable': 1, 'cup': 1, 'wine glass': 1}	1
{'tie': 3, 'person': 3, 'book': 1, 'diningtable': 1, 'chair': 1, 'cup': 1, 'wine glass': 1}	1
{'person': 3, 'tie': 2, 'book': 1, 'diningtable': 1, 'chair': 1, 'wine glass': 1}	1
{'person': 3, 'tie': 2, 'book': 1, 'chair': 1, 'cup': 1, 'wine glass': 1}	1
{'person': 3, 'book': 2, 'tie': 2, 'wine glass': 1}	1
{'person': 3, 'book': 1, 'cup': 1, 'wine glass': 1, 'tie': 1}	1
Total geral	**87**

Figura 17 – Relação de objetos detectados para todos os 87 *frames* processados do vídeo 1, por ordem de ocorrência. Fonte: Própria (2022).

Com relação a repetição de resultados de classificação, o mesmo pôde ser observado no vídeo 2 (Figura 18). Onde 69% dos *frames* processados tiveram o mesmo resultado, o que pode indicar que, dentro do intervalo de tempo analisado, a câmera se manteve fixa para um mesmo quadro; a troca de câmeras pôde ser confirmada com as classificações onde mais de 10 pessoas foram detectadas, indicando a filmagem do auditório.

Objetos Detectados	**Nº Frames**
{'person': 2, 'tie': 1}	45
{'person': 9, 'tie': 4, 'chair': 3, 'laptop': 1, 'cup': 1}	1
{'person': 9, 'tie': 3, 'laptop': 2, 'cup': 1}	1
{'person': 8, 'tie': 4, 'chair': 1, 'cup': 1}	1
{'person': 8, 'tie': 3, 'laptop': 2, 'chair': 2, 'cell phone': 1, 'cup': 1}	1
{'person': 8, 'tie': 3, 'chair': 2, 'laptop': 1, 'cup': 1}	1
{'person': 8, 'laptop': 2, 'chair': 2, 'tie': 2, 'cup': 1}	1
{'person': 8, 'chair': 2, 'cup': 1, 'tie': 1}	1
{'person': 7, 'tie': 3, 'laptop': 2, 'chair': 1, 'cup': 1}	1
{'person': 7, 'laptop': 1, 'chair': 1, 'cup': 1, 'tie': 1}	1
{'person': 6, 'tie': 3, 'laptop': 1, 'chair': 1}	1
{'person': 3, 'tie': 1}	1
{'person': 2, 'cell phone': 1, 'tie': 1}	1
{'person': 15, 'tie': 5, 'chair': 4, 'laptop': 1}	1
{'person': 14, 'chair': 7, 'tie': 4, 'book': 1, 'laptop': 1}	1
{'person': 14, 'chair': 6, 'tie': 4, 'laptop': 1}	1
{'person': 14, 'chair': 5, 'tie': 5, 'laptop': 1}	1
{'person': 12, 'tie': 7, 'chair': 5, 'laptop': 1}	1
{'person': 12, 'chair': 4, 'tie': 4, 'laptop': 1}	1
{'person': 11, 'tie': 5, 'chair': 4, 'laptop': 1}	1
{'person': 11, 'chair': 5, 'tie': 4, 'laptop': 1}	1
Total geral	**65**

Figura 18 – Relação de objetos detectados para todos os 65 *frames* processados do vídeo 2, por ordem de ocorrência. Fonte: Própria (2022).

A partir da relação de objetos detectados nos *frames* do vídeo 3, pôde-se confirmar que o cenário do vídeo se ambienta em uma área a céu abertom uma vez que houve a classificação de automóveis. Além disso, conforme pode ser visto na Figura 19, para um dos *frames* processado, houve a classificação de um ônibus (*bus*) porém não há ônibus na cena e sim a ambulância.

objetos detectados	Nº Frames
{'person': 3, 'tie': 1}	14
{'person': 5, 'tie': 3}	6
{'person': 4, 'tie': 1}	6
{'person': 5, 'tie': 2}	5
{'person': 5, 'tie': 1}	4
{'person': 3, 'tie': 1, 'car': 1}	4
{'person': 4, 'tie': 3}	3
{'person': 3, 'tie': 1, 'truck': 1}	3
{'person': 5, 'truck': 2, 'tie': 1}	2
{'person': 5, 'tie': 4}	2
{'person': 5, 'tie': 1, 'truck': 1}	2
{'person': 4, 'tie': 2}	2
{'person': 3, 'tie': 2, 'car': 1}	2
{'person': 2}	2
{'person': 2, 'tie': 1}	2
{'truck': 3, 'person': 3, 'tie': 2}	1
{'person': 5}	1
{'person': 5, 'truck': 2, 'tie': 1, 'car': 1}	1
{'person': 5, 'tie': 2, 'truck': 1}	1
{'person': 5, 'tie': 2, 'car': 1}	1
{'person': 4, 'tie': 3, 'car': 2}	1
{'person': 4, 'tie': 1, 'truck': 1}	1
{'person': 4, 'tie': 1, 'car': 1}	1
{'person': 3, 'tie': 1, 'bus': 1}	1
Total geral	**68**

Figura 19 – Relação de objetos detectados para todos os 68 *frames* processados do vídeo 3, por ordem de ocorrência. Fonte: Própria (2022).

Quanto ao objetos detectados no vídeo 4 (Figura 20), o que chamou a atenção foram as classificações vazias para 4 *frames*. Ou seja, nenhum objeto foi detectado nessas imagens. Olhando para esses quatro frames (Figura 21), isso se justifica para as imagens (a) e (b), onde há a presença de um objeto originário da comunidade indígena, e esse objeto não está presente no *dataset* utilizado para treinamento do modelo utilizado. E quanto a imagem (d) realmente não há presença de nenhum objeto, somente textos e não é objeto deste trabalho fazer reconhecimento de textos.

Além disso, percebe-se também a detecção de objetos que não estavam presentes nas cenas processadas. Como, por exemplo, *bear*, *elephant*, *cake* e *baseball bat*.

objetos detectados	Nº Frames
{'book': 4, 'person': 1}	29
{'book': 3, 'person': 1}	29
{'person': 1}	16
{'tvmonitor': 1, 'person': 1}	14
{'book': 3, 'clock': 1, 'person': 1}	7
{'book': 2, 'person': 1}	7
{'tvmonitor': 2, 'book': 1, 'keyboard': 1, 'laptop': 1, 'p	6
{'person': 3}	6
{'tvmonitor': 2, 'book': 1, 'keyboard': 1, 'person': 1}	5
{'tvmonitor': 3, 'book': 1, 'keyboard': 1, 'person': 1}	4
{'person': 2}	4
{'book': 4, 'clock': 1, 'person': 1}	4
	4
{'tvmonitor': 3, 'book': 1, 'keyboard': 1, 'laptop': 1, 'p	3
{'book': 5, 'person': 1}	3
{'person': 6}	2
{'person': 4}	2
{'book': 3, 'remote': 1, 'person': 1}	2
{'person': 9}	1
{'person': 8, 'elephant': 1}	1
{'person': 7}	1
{'person': 3, 'cake': 1}	1
{'person': 25, 'backpack': 1}	1
{'person': 24, 'suitcase': 1, 'backpack': 1}	1
{'person': 23}	1
{'person': 21, 'suitcase': 1}	1
{'person': 2, 'traffic light': 1}	1
{'person': 13, 'baseball bat': 1, 'sports ball': 1}	1
{'person': 1, 'cat': 1}	1
{'dog': 1, 'person': 1}	1
{'dog': 1, 'horse': 1, 'person': 1}	1
{'book': 6, 'person': 1}	1
{'book': 2, 'clock': 1, 'person': 1}	1
{'bear': 2}	1
{'bear': 1}	1
Total geral	**164**

Figura 20 – Classes dos objetos detectados para os *frames* do vídeo 4, por ordem de ocorrência. Fonte: Própria (2022).

(a) img22.jpg.

(b) img23.jpg.

(c) img34.jpg.

(d) img164.jpg.

Figura 21 – *Frames* do vídeo 4 onde nenhum objeto foi detectado pelo modelo, seja por motivo de o objeto presente no frame não fazer parte dos objetos conhecidos pelo modelo em seu conjunto de treinamento (a) e (b), ou por realmente não haver nenhum objeto no frame, somente texto (d) Fonte: Própria (2022).

6.2.3 Detecção automática de ação realizada pelas pessoas detectadas no vídeo

Com os arquivos *.mp4* referentes aos 4 vídeos da Tabela 4, o processo de detecção de ação (Seção 5.3), foi executado e a extração das características foi feita. Essas foram escritas em um arquivo *.csv* e o qual foi armazenado no banco de dados. Na Figura 22 há um exemplo de classificação para um dos *frames* dos vídeos listados.

(a) Ações detectadas no *frame* do vídeo 1

(b) Ações detectadas no *frame* do vídeo 2

(c) Ações detectadas no *frame* do vídeo 3

(d) Ações detectadas no *frame* do vídeo 4

Figura 22 – Ilustração das ações detectadas por pessoas presentes em um dos *frames* dos 4 vídeos. Fonte: Própria (2022).

Manipulando o arquivo salvo com as classes das ações detectadas, para o intervalo definido para o vídeo 1, as ações listadas na Tabela 5 foram as mais recorrentes ao longo das cenas. Com esse resultado, percebeu-se que o modelo detectou as ações 'carry/hold (an object)' e 'touch (an object)', as quais podem estar relacionadas com os momentos em que as pessoas, durante o vídeo, fazem movimentos com o papel, ou caneta, ou óculos na mão. Nesse caso, como o modelo usado não consegue detectar qual objeto que está relacionado com esta ação, seria necessário pegar a referência do frame referente à aquela predições e confirmar quais objetos estavam sendo manuseados naquele momento.

Tabela 5 – Ações detectadas ao longo do Vídeo 1. Fonte: Própria (2022).

Ações detectadas
answer phone
carry/hold (an object)
drink
listen to (a person)
sit
stand
talk to (e.g., self, a person, a group)
touch (an object)
watch (a person)

As classes listadas na Tabela 6 foram as ações detectadas ao longo do intervalo do vídeo 2, processado pela ferramente proposta.

Tabela 6 – Ações detectadas ao longo do Vídeo 2. Fonte: Própria (2022).

Ações detectadas
carry/hold (an object)
listen to (a person)
sit
stand
talk to (e.g., self, a person, a group)
touch (an object)
watch (a person)

Já para o vídeo 3, as ações listadas na Tabela 7 foram detectadas ao longo do intervalo processado. O que chamou atenção foi a ocorrência de detecção da classe 'sit', o que não era pra ter acontecido, uma vez que em nenhum momento, durante o intervalo de tempo do vídeo processado, foi filmada uma pessoa sentada.

Tabela 7 – Ações detectadas ao longo do Vídeo 3. Fonte: Própria (2022).

Ações detectadas
listen to (a person)
sit
stand
talk to (e.g., self, a person, a group)
watch (a person)

Na Tabela 8 encontram-se as classes detectadas como ações reproduzidas ao longo do intervalo do vídeo 4 processado pela ferramenta. Comparadas com os resultados dos vídeos anteriores, percebeu-se a ocorrência de duas detecções diferentes. São elas: bend/bow (at the waist) e lie/sleep.

Tabela 8 – Ações detectadas ao longo do Vídeo 4. Fonte: Própria (2022).

Ações detectadas
bend/bow (at the waist)
carry/hold (an object)
lie/sleep
listen to (a person)
sit
stand
talk to (e.g., self, a person, a group)
watch (a person)

6.3 Relação das características extraídas com os elementos da narrativa audiovisuais

Diante da validação dos processos de executados pela ferramenta proposta para extração de características, através dos resultados obtidos dos experimentos realizados e descritos nos tópicos anteriores, busca-se relacionar as características extraídas, da parte visual do vídeo, com alguma das características que compõem os elementos de uma narrativa audiovisual. Para isso, foi levado em consideração o que esta descrito

na Seção 5.4 deste trabalho. Onde consta que uma narrativa é composta por alguns principais campos que são caracterizados por elementos que estão presentes em vídeos.

Sendo assim, tendo a Tabela 3 como referência, as Tabelas 9, 10, 11, 12 foram montadas para ilustrar como as características extraídas automaticamente, neste trabalho, podem ser relacionadas e utilizadas como elementos que caracterizam o **espaço/ambiente** e as **Ações/Movimentos** presente na narrativa audiovisual dos vídeos utilizados nos experimentos.

Tabela 9 – Relação das características extraídas do vídeo 1, com os elementos da narrativa audiovisual. Fonte: Própria (2022).

	ESPAÇO/AMBIENTE	**PERSONAGEM/PESSOA E AÇÃO**
Elementos que compõem a narrativa	**Objetos**, Cena e Cenário	Interação objeto x pessoa
caracteristicas extraídas automáticamente	book, chair, cup, diningtable, laptop, person, tie, wine glass.	answer phone, carry/hold (an object), drink, listen to (a person), sit, stand, talk to (e.g., self, a person, a group), touch (an object), watch (a person).

Tabela 10 – Relação das características extraídas do vídeo 2, com os elementos da narrativa audiovisual. Fonte: Própria (2022).

	ESPAÇO/AMBIENTE	**PERSONAGEM/PESSOA E AÇÃO**
Elementos que compõem a narrativa	**Objetos**, Cena e Cenário	Interação objeto x pessoa
caracteristicas extraídas automáticamente	cell phone, chair, cup, laptop, person, tie.	carry/hold (an object), listen to (a person), sit, stand, talk to (e.g., self, a person, a group), touch (an object), watch (a person).

Tabela 11 – Relação das características extraídas do vídeo 3, com os elementos da narrativa audiovisual. Fonte: Própria (2022).

	ESPAÇO/AMBIENTE	**PERSONAGEM/PESSOA E AÇÃO**
Elementos que compõem a narrativa	**Objetos**, Cena e Cenário	Interação objeto x pessoa
caracteristicas extraídas automáticamente	book, bus, car, person, truck, tie.	listen to (a person), sit, stand, talk to (e.g., self, a person, a group), watch (a person).

Tabela 12 – Relação das características extraídas do vídeo 4, com os elementos da narrativa audiovisual. Fonte: Própria (2022).

	ESPAÇO/AMBIENTE	**PERSONAGEM/PESSOA E AÇÃO**
Elementos que compõem a narrativa	**Objetos**, Cena e Cenário	Interação objeto x pessoa
caracteristicas extraídas automáticamente	backpack, bear, book, cake, cat, clock, cup, dog, elephant, horse, keyboard, laptop, person, remote, baseball bat, sport ball, suitcase, tie, traffic light, tvmonitor.	bend/bow (at the waist), carry/hold (an object), lie/sleep, listen to (a person), sit, stand, talk to (e.g., self, a person, a group), watch (a person).

6.3.1 Espaço/Ambiente

Para o estudo do Espaço/Ambiente da narrativa audiovisual de um dos 4 vídeos (Tabela 4), um pesquisador utilizando a ferramenta proposta poderá extrair

automaticamente os objetos daqueles vídeos (e.g. Figuras 17, 18, 19, 20). Além disso, conforme afirmado por Cirino (2021), esses mesmos objetos fazem parte dos recursos visuais que caracterizam elementos narrativos como: composição de cena, cenário, movimentação de câmera e enquadramento.

6.3.2 Ação / Movimento

Para o estudo de ações e movimentos realizados por personagens da narrativa audiovisual, de um dos 4 vídeos usados na parte experimental deste trabalho, um pesquisador, utilizando a ferramenta proposta, poderá extrair, automaticamente, as principais ações reproduzidas por pessoas detectadas nos vídeos. Exemplos dessas ações/movimentos retornadas foram vistos nas Tabelas 5, 6, 7, 8.

7 Considerações finais

Neste trabalho foi proposto o desenvolvimento de uma ferramenta, cujo objetivo principal era de automatizar o processo de identificação e extração de características visuais presentes em vídeos que foram publicado na plataforma YouTube. A qual pode ser integrada no processo de caracterização e análise de narrativas audiovisuais, uma vez que as características extraídas pela ferramenta se relacionam com os principais elementos que compõem os recursos visuais da narrativa.

Para o desenvolvimento da ferramenta, a qual foi realizada utilizando a linguagem Python, fora realizada a integração e, posterior, aplicação de dois modelos pré-treinados, de *deep learning*, os quais executaram a tarefa de detecção de objetos (YoloV4) e detecção de ação (SlowFast), para os vídeos processados. Durante a execução do processo de classificação das características desejadas, houve a predição não acuradas, o que já era esperado, uma vez que os aquivos de treinamentos utilizados para ambos os modelos têm media de precisão correta menor do que 75%, para predições corretas. Embora esses resultados sejam preliminares, podendo ser melhorados com a substituição dos modelos pré-treinados por outros que obtiverem maiores taxas de precisão.

A validação da arquitetura proposta (apresentada na seção 4.2) se deu através dos resultados obtidos na seção 6.3, deste trabalho, onde foi feita a relação das características extraídas da parte visual do vídeo com os elementos da narrava audiovisual, seguindo o que foi definido na seção 2.3. Sendo assim, os resultados dos experimentos, realizados para os 4 vídeos listados na Tabela 4, afirmam a eficácia da metodologia proposta para a extração das características definidas.

Ou seja, através dos resultados obtidos da fase de extração de dados, em conjunto com a detecção de objetos e detecção automática de ação/atividades, pode-se afirmar que a ferramenta desenvolvida faz a extração de características que se relacionam com narrativas audiovisuais. O que possibilita a sua integração no processo de análise de narrativas desse tipo, automatizando e, consequentemente, otimizando o processo de extração de características de vídeos.

Como sugestões para trabalhos futuros:

- Experimentar a ferramenta desenvolvida com uso de vídeos de diferentes categorias, com o objetivo de identificar possíveis melhorias na lógica do código atual. O que tornaria a ferramenta mais dinâmica e mais otimizada com relação ao tempo de execução de cada processo.

- Na fase de detecção de objetos, foi observado que, mesmo que o modelo utilizado tenha sido treinado com 80 classes diferentes, há objetos presentes nos vídeos processados que não foram classificados, por não fazer parte daquele conjunto de treinamento (*e.g.* papel, caneta, bandeira, óculos, microfone). Com isso, o treinamento de um novo modelo do YoloV4, incluindo essas e outras classes de objetos, que são frequentes em vídeos da categoria utilizada neste trabalho, é uma sugestão para trabalho futuro;

- Foi observado nos resultados da detecção de ação, que quando uma pessoa detectada faz movimentos segurando algum objeto em suas mãos, o modelo utilizado não faz a classificação desse objeto, apenas prediz se a pessoa ta carregando ou tocando um objeto (não identificado). Portanto, a realização do estudo e testes para validar a possibilidade do aprimoramento do processo de detecção de ação, a partir do manuseio de objetos, é uma sugestão de trabalho futuro que pode contribuir para a área do estudo de classificação de vídeos;

- Analisar a relação das características extraídas, pela ferramenta desenvolvida, com o estudo das características que indicam o estado emocional das pessoas detectadas nos vídeos.

  – Também motivado pelo item acima, o autor deste trabalho submeteu ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas uma proposta de pesquisa de Mestrado, que teve como título "Reconhecimento automático de emoções em vídeo a partir da extração de características audiovisuais". Proposta essa que foi aprovada pelo EDITAL N 055/2021 PROPESP/UFAM; cujo início da pesquisa será a partir do mês de outubro de 2022.

Referências

BISONG, E. Google colaboratory. In: ____. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley, CA: Apress, 2019. p. 59–64. ISBN 978-1-4842-4470-8. Disponível em: <https://doi.org/10.1007/978-1-4842-4470-8_7>. 38

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 16, 20, 22, 25, 26

CARTA, S. et al. Vstar: Visual semantic thumbnails and tags revitalization. *Expert Systems with Applications*, Elsevier, p. 116375, 2022. 27

CHIOU, M.-J.; ZIMMERMANN, R.; FENG, J. Visual relationship detection with visual-linguistic knowledge from multimodal representations. *IEEE Access*, IEEE, v. 9, p. 50441–50451, 2021. 22

CIRINO, C. *A Desinformação sobre a Amazônia no Youtube: Padrões de narrativa com o uso de Inteligência Artificial*. [S.l.], 2021. 15, 16, 17, 23, 53

CIRINO, C. et al. A amazônia e polarização política no youtube: Representação de narrativas com o uso de sistema de inteligência artificial. In: *3ž Seminário Internacional América Latina - SIALAT, 2021*. [S.l.: s.n.], 2021. p. 2511–2529. 15, 16, 25, 38

CISCO, U. Cisco annual internet report (2018–2023) white paper. *Online](accessed March 26, 2021) https://www. cisco. com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/whitepaper-c11-741490. html*, 2020. 15

FAN, H. et al. Pytorchvideo: A deep learning library for video understanding. In: *Proceedings of the 29th ACM International Conference on Multimedia*. [S.l.: s.n.], 2021. p. 3783–3786. 16, 22, 23, 25, 26

FEICHTENHOFER, C. et al. Slowfast networks for video recognition. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2019. p. 6202–6211. 16, 22, 25, 26

GARG, S. et al. Semantics for robotic mapping, perception and interaction: A survey. *Foundations and Trends® in Robotics*, Now Publishers, Inc., v. 8, n. 1–2, p. 1–224, 2020. 21, 22, 25

GKIOXARI, G. et al. Detecting and recognizing human-object interactions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 8359–8367. 21

GUIMARÃES, V. de S. *Identificação de Características de Aspectos Emocionais Associados a Elementos de Narrativas Audiovisuais*. [S.l.], 2022. 9, 12, 23, 24, 25, 31

MAO, F. et al. Hierarchical video frame sequence representation with deep convolutional graph network. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. [S.l.: s.n.], 2018. p. 0–0. 22

MIHAJLOVIĆ, S. et al. The use of python in the field of artifical intelligence. 2020. 16

MOTTA, L. G. Análise pragmática da narrativa jornalística. In: INTERCOM. *Congresso Brasileiro de Ciências da Comunicação*. [S.l.], 2005. v. 28, p. 05–09. 37

OPENCV. *Open Source Computer Vision Library*. 2015. Disponível em: <https://github.com/opencv/opencv/tree/4.5.5>. 31

PAFY. *pafy 0.5.5*. Python Software Foundation, 2019. Disponível em: <https://pypi.org/project/pafy/>. 31

PYTORCHVIDEO. *Running a pre-trained PyTorchVideo classification model using Torch Hub*. Facebook Open Source, 2021. Disponível em: <https://pytorchvideo.org/docs/tutorial_torchhub_detection_inference>. 35, 60

REBACK, J. et al. *pandas-dev/pandas: Pandas 1.4.2*. Zenodo, 2022. Disponível em: <https://doi.org/10.5281/zenodo.6408044>. 31

REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. *arXiv*, 2018. 9, 19, 20, 21

RUSSEL, S.; NORVIG, P. et al. *Artificial intelligence: a modern approach*. [S.l.]: Pearson Education Limited London, 2020. v. 4. 750 p. 19

RUSSEL, S.; NORVIG, P. et al. *Artificial intelligence: a modern approach*. [S.l.]: Pearson Education Limited London, 2020. v. 4. 889 – 901 p. 19

WU, Y. et al. *Detectron2*. 2019. <https://github.com/facebookresearch/detectron2>. 25, 26

YOHANANDAN, S. P. S. *mAP (mean Average Precision) might confuse you!* Towards Data Science, 2020. Disponível em: <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>. 33

YOUTUBE. *YouTube Data API*. Google Developers, 2022. Disponível em: <https://developers.google.com/youtube/v3/docs>. 31

APÊNDICE A – Principais funções definidas

A seguir encontra-se algumas das principais funções definidas em Python, que executam parte dos principais processos presentes na arquitetura proposta:

- Funções de Acesso ao Vídeo:

```
#retorna objetos com os metadados do vídeo input
def get_video_pafy(videoId):
    url = "https://www.youtube.com/watch?v=" + videoId
    video = pafy.new(url)
    best  = video.getbest(preftype="mp4")
    return video, best

def download_video(videoId):
    url = "https://www.youtube.com/watch?v=" + videoId
    video = pafy.new(url)
    best  = video.getbest(preftype="mp4")
    filename = best.download(filepath="videoInput." + best.extension)
    return filename
```

- Função de Extração de Frames:

```
def extract_frames_openCv(videoUrl, tempoInicial,
    tempoFinal, folderPath):
    indexFrame = 0
    countFrame = 1
    tempos = []
    nomeFrames = []
    dataVideo = cv2.VideoCapture(videoUrl)
    fps = int(dataVideo.get(cv2.CAP_PROP_FPS))
    print("FPS:", fps)
```

```
    while(dataVideo.isOpened()):
        ret, frame = dataVideo.read()
        #calculando o tempo que ocorreu:
        seconds = int(indexFrame/fps)
        frame_time = str(datetime.timedelta(seconds=seconds))
        if(seconds >= tempoInicial):
             ##estipula um tempo máx para leitura de frames
            if seconds >= tempoFinal: break
            if ret == False: break
            if indexFrame%fps == 0: #processa 1 frame por segundo
                nome = 'img' + str(countFrame) + '.jpg'
                tempos.append(frame_time)
                nomeFrames.append(nome)
                cv2.imwrite(folderPath + "/" + nome, frame)
                countFrame += 1
        indexFrame += 1
    dataVideo.release()
    cv2.destroyAllWindows()
    print("finished!")

    #relação entre o frame extraido com o seu tempo de
    acontecimento no vídeo
    df_framesExtraidos = {
        'tempo' : tempos,
        'frame' : nomeFrames}
    columns = ['tempo','frame']
    df_framesExtraidos = pd.DataFrame(data=df_framesExtraidos,
    columns=columns)
    return df_framesExtraidos
```

- Comando de execução do processo de predição do Yolov4:

#Here is an example of saving the multiple image detections to a

```
#json file.
!./darknet detector test cfg/coco.data cfg/yolov4.cfg
yolov4.weights -ext_output -dont_show -out
/content/ObjDetcResult.json < /content/images.txt
```

- Função para detecção de pessoas para predição de ação. Fonte: (PYTORCHVIDEO, 2021):

```
# This method takes in an image and generates the
#bounding boxes for people in the image.
def get_person_bboxes(inp_img, predictor):
    predictions = predictor(inp_img.cpu().detach().numpy())
    ['instances'].to('cpu')
    boxes = predictions.pred_boxes if predictions.has("pred_boxes")
    else None
    scores = predictions.scores if predictions.has("scores")
    else None
    classes = np.array(predictions.pred_classes.tolist()
    if predictions.has("pred_classes") else None)
    predicted_boxes = boxes[np.logical_and(classes==0,
    scores>0.75 )].tensor.cpu() # only person
    return predicted_boxes
```

- Função que salva os resultados da detecção de ação:

```
def save_frame_Wpred(frame, index, outputPath, fps):
    nome = 'img' + str(index) + '.jpg'
    cv2.imwrite(folderPath + "/" + nome, frame)
    #print("frame {} saved".format(nome))

videoOutput = cv2.VideoWriter(vide_save_path,cv2.VideoWriter_fourcc
(*'DIVX'), 7, (width,height))
index=0
for image in gif_imgs:
```

```
10      img = (255*image).astype(np.uint8)
11      img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
12      save_frame_Wpred(img, index, folderPath,1)
13      videoOutput.write(img)
14      index+=1
15  videoOutput.release()
```