

ANÁLISE DE SENTIMENTOS EM IMAGEM

APRESENTADORES: CARLOS EDUARDO E LUIZ SACRAMENTO

OBJETIVO

ANÁLISAR E DESCREVER OS TESTES DE ANÁLISE DE SENTIMENTO EM IMAGEM.

PERGUNTA NORTEADORA

QUAIS OS PRINCIPAIS DESAFIOS ENCONTRADOS NO
ALGORITMO DE ANÁLISE DE SENTIMENTO EM
IMAGEM, UTILIZADO NESSE EXPERIMENTO?

PRINCIPAIS BIBLIOTECAS UTILIZADAS

OPENCV
PYTEST
TENSORFLOW

DIFERENÇA ENTRE BIBLIÓTECAS

OPENCV E PILLOW

A DIFERENÇA BÁSICA ENTRE A IMAGEM DO OPENCV E O PIL É QUE O OPENCV SEGUE A CONVENÇÃO DE CORES BGR E PIL SEGUE A CONVENÇÃO DE CORES RGB E O MÉTODO DE CONVERSÃO SERÁ BASEADO NESTA DIFERENÇA.

PYTEST E UNITTEST

	Unittest	Pytest
Linguagem de Programação	Python	Python
Informações gerais	A estrutura de teste unittest é a estrutura de estilo xUnit do Python. É um módulo padrão que acompanha o Python e oferece suporte à automação e agregação de testes e configuração comum e código de desligamento para eles.	Pytest é uma poderosa estrutura de teste Python que pode testar todos os níveis de software. É considerado por muitos como o melhor framework de teste em Python, com muitos projetos na internet mudando para ele de outros frameworks, incluindo Mozilla e Dropbox. Isso se deve a seus muitos recursos poderosos, como reescrita 'assert', um modelo de plug-in de terceiros e um modelo de fixação poderoso, mas simples.
Teste unitário	Sim	Não
Client-side	Sim, seu teste pode ser aplicado no front-end	Sim, seu teste pode ser aplicado no front-end
Server-side	Sim, seu teste pode ser aplicado no back-end	Sim, seu teste pode ser aplicado no back-end

fonte: https://knapsackpro.com/testing_frameworks/difference_between/unittest/vs/pytest

TENSORFLOW

"*TENSORFLOW*" É UMA BIBLIOTECA DE CÓDIGO ABERTO CRIADA PARA APRENDIZADO DE MÁQUINA, COMPUTAÇÃO NUMÉRICA E MUITAS OUTRAS TAREFAS. FOI DESENVOLVIDO PELO GOOGLE EM 2015 E RAPIDAMENTE SE TORNOU UMA DAS PRINCIPAIS FERRAMENTAS PARA MACHINE LEARNING E DEEP LEARNING.

QUALQUER PESSOA PODE UTILIZAR O TENSORFLOW GRATUITAMENTE, E O ESTILO DE PROGRAMAÇÃO EM TENSORFLOW É MUITO SEMELHANTE AO PYTHON."

fonte: <https://didatica.tech/o-que-e-tensorflow-para-que-serve/>

METODOLOGIA

A METODOLOGIA FOI DIVIDIDA EM 5 ETAPAS, SENDO ESSAS SEGUIDAS DE MANEIRA SEQUENCIADAS, OU SEJA, UMA ETAPA EXECUTADA OBRIGATORIAMENTE APÓS A OUTRA.

NO PLANEJAMENTO É USAR UM BANCO COM 35.887 IMAGENS(FER2013, É OPENSOURCE CRIADO PELO PIERRE-LUCCARRIER E AARON COURVILLE)

ETAPA 1

IMPORTANDO AS BIBIOTECAS

```
1 import cv2
2 import numpy as np
3 import pandas as pd
```

Todas essas bibliotecas, caso já não estejam instaladas, podem ser adquiridas via PIP. A biblioteca pandas é usada para se trabalhar primordialmente com análise de dados numéricos, agora segue abaixo outras bibliotecas também utilizadas.

A OpenCV 4.6.0 é a biblioteca que será utilizada para o processamento de imagens, nela que se encontra o método cv, usado para a leitura das imagens.

Em seguida, o Numpy 1.22.0 também será instalado, essa biblioteca para trabalhar com manipulações matemáticas e matrizes, algo bem utilizado na hora de verificar o posicionamento das fotos.

ETAPA 2

CONECTANDO COM O DRIVE E ACESSANDO OS ARQUIVOS

```
1 from google.colab import drive  
2 drive.mount('/content/gdrive')
```

Nessa etapa, é feita a conexão entre o Colab e a conta do drive utilizada para banco de imagens e o arquivo com os modelos usados para detecção de faces.

ETAPA 3

DETECÇÃO DE FACES

```
face_cascade = cv2.CascadeClassifier('/content/drive/MyDrive/  
documentos_python/haarcascade_frontalface_default.xml')
```

O classificador de imagem utilizado foi o cascadeClassifier que trabalha com a classificação de imagens em RGB. Também é opensource esse arquivo para classificação de imagem e está no link abaixo do Github

fonte: https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml

ETAPA 3

DETECÇÃO DE FACES

```
original = imagem.copy()  
gray = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)  
cv2_imshow(gray)
```

Esse é o algoritmo descolore a foto para escala cinza e detecta a face, o qual é a preparação para a próxima etapa

ETAPA 3

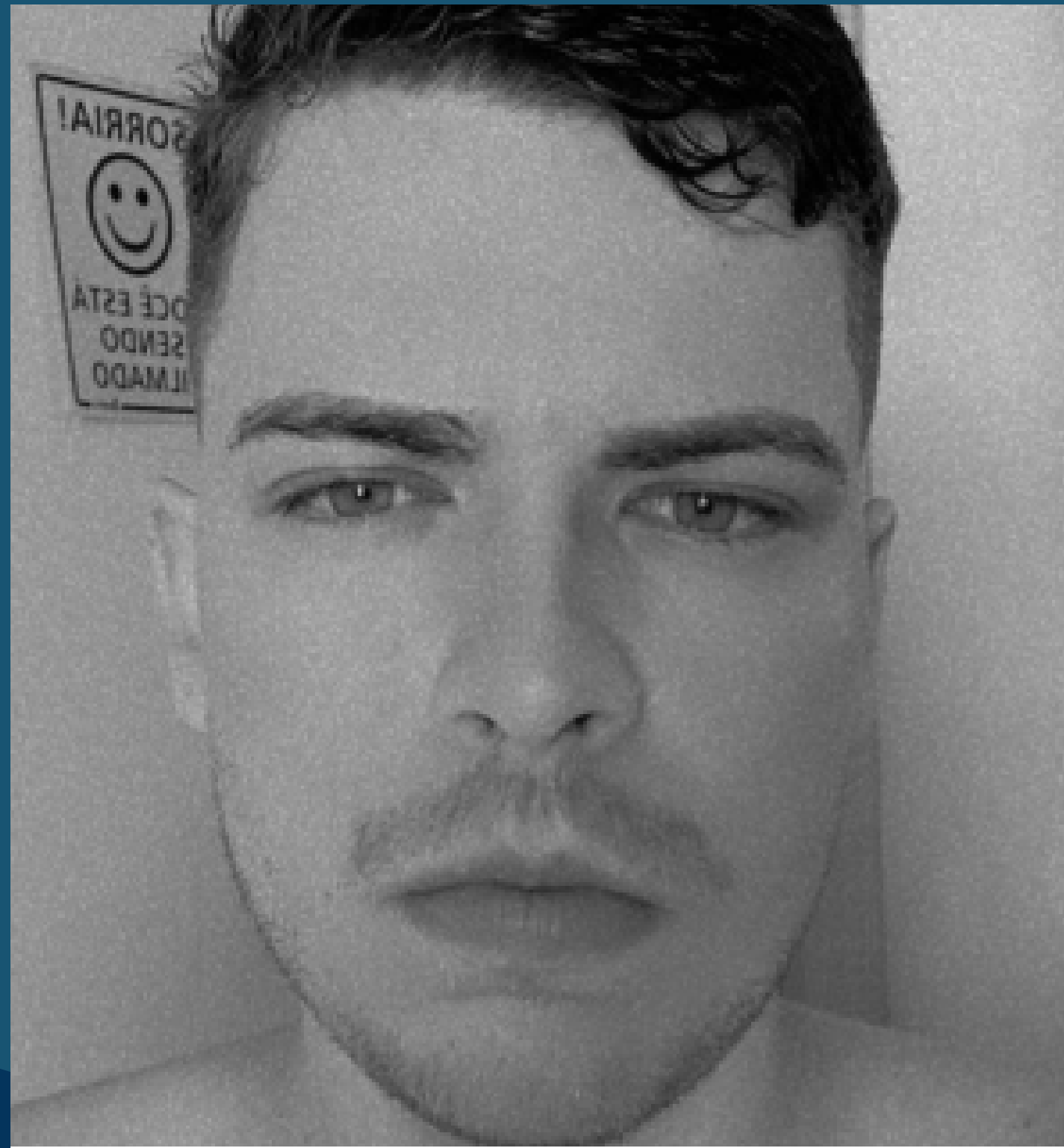
Antes
DETECÇÃO DE FACES E MANIPULAÇÃO DE IMAGENS



ETAPA 3

DETECÇÃO DE FACES

Depois



ETAPA 4

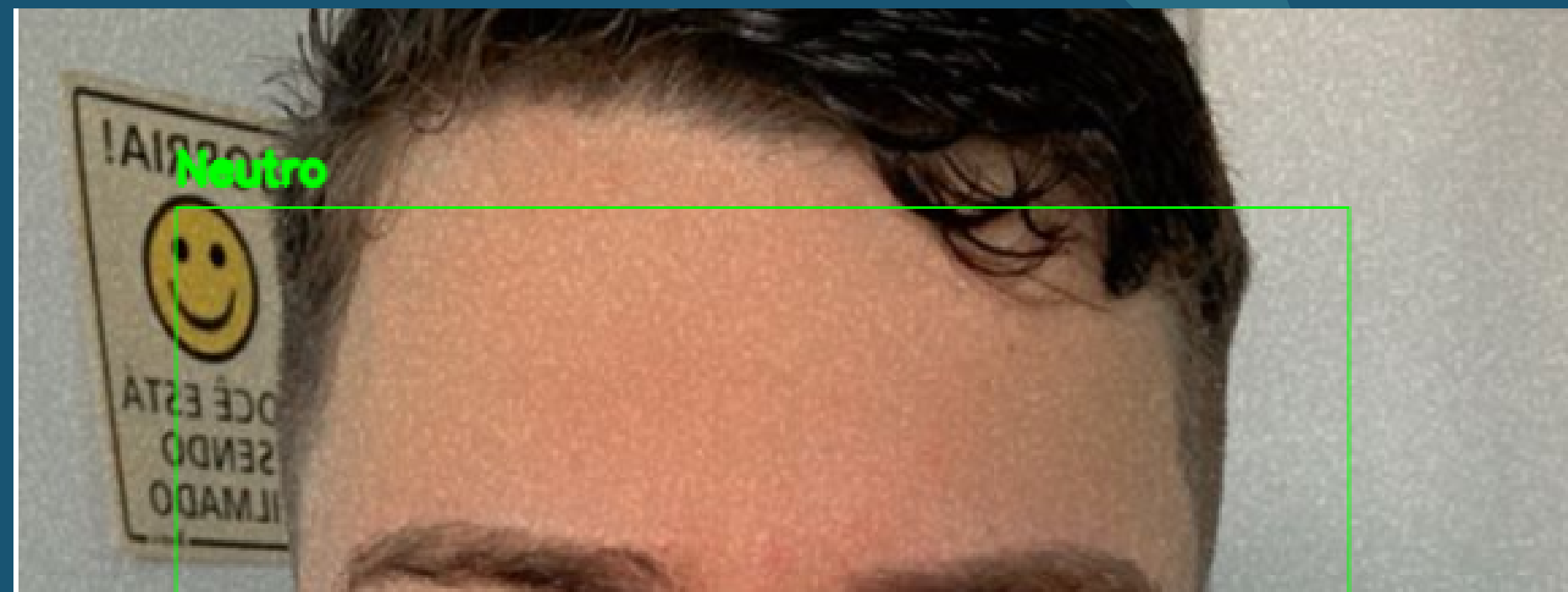
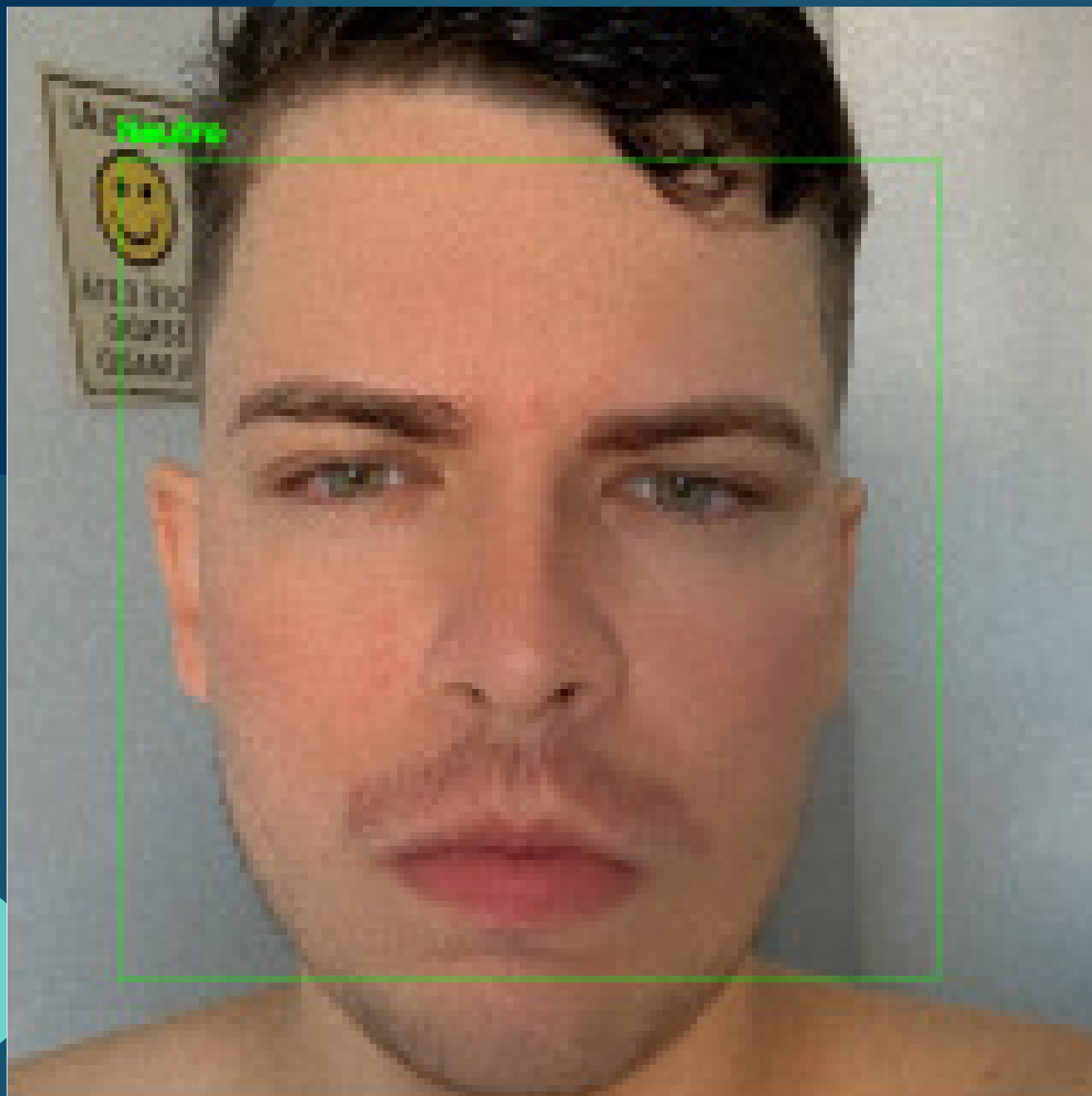
ANÁLISE DE SENTIMENTOS

Por fim, é definido o sentimento presente em cada uma das imagens, com o código abaixo

```
2  
3 for (x,y,w,h) in faces:  
4     cv2.rectangle(original, (x,y), (x+y,y+h), (0,255,0), 1)  
5 cv2_imshow(original)
```

ETAPA 4

ANÁLISE DE SENTIMENTOS



ETAPA 5

REALIZAR O EXPERIMENTO

O próximo passo é a realização do teste unitário, comentar as saídas encontradas ao analisar tais fotos e se realmente o algoritmo cumpre o prometido.

REFERENCIAL TEÓRICO

NOS INSPIRAMOS EM UM ARTIGO QUE FALA SOBRE RECONHECIMENTO DE EXPRESSÃO FACIAL COM DEEP LEARNING. NELE É APRESENTADO MODELOS COMPUTACIONAIS FOI USADA ESSA BIBLIOTECA CHAMADA FER. DURANTE ESSE EXPERIMENTO, FOI-SE UTILIZADO JUSTAMENTE ESSE BANCO DE IMAGENS OPENSOURCE.

REFERENCIAL TEÓRICO

SEGUNDO A REFERÊNCIA DO ARTIGO " RECONHECIMENTO DE EXPRESSÕES FACIAIS COM DEEP LEARNING"

"FER2013 É UM CONJUNTO DE DADOS BEM ESTUDADO E TEM SIDO USADO EM COMPETIÇÕES ICML E VÁRIOS TRABALHOS DE PESQUISA. É UM DOS CONJUNTOS DE DADOS MAIS DESAFIADORES, COM PRECISÃO DE NÍVEL HUMANO[...]"(P-2, BAI, CHARLES; KHANZADA, AMIL; CELEP-CIKAY, FERHAT)

fonte: <https://arxiv.org/abs/2004.11823>

OBJETIVOS ATINGIDOS?

1 - TESTES

2 - ANÁLISE CRÍTICA DO MODELO UTILIZADO

SIM, A SEGUIR TAIS OBJETIVOS SERÃO MOSTRADOS
NOS RESULTADOS:

RESULTADOS E DISCUSSÕES

DENTRE OS RESULTADOS OBTIDOS, O PRIMEIRO A SER CITADO É O TESTE UNITÁRIO, O QUAL FOI UTILIZADO PRIMORDIALMENTE NO MOMENTO DA DIGITALIZAÇÃO DE IMAGENS.

ESTRUTURA DO TESTE

```
14     try:
15         if len(imagem)>0:
16             # Do something if
17             logging.info(" Successfully loaded file")
18             return 1
19
20     except FileNotFoundError:
21         # File doesn't exist
22         logging.info(" File not found/ Incorrect file path")
23         return 2
24
25     else:
26         # Return error loading file
27         logging.info(" File load failed")
28         return 3
29
30
31 class TestFileloader:
32     def testthis(self):
33         status = teste('Material/testes/teste02.jpg')
34         assert status ==2
```


NESSA TESTE FICOU VISÍVEL A MÁ PRÁTICA DE PROGRAMAÇÃO EM NÃO DEFINIR UMA VARIÁVEL ANTES DE USÁ-LA

```
22
23 self = <test_file_load.TestFileloader object at 0x7f87b0798d10>
24
25     def testthis(self):
26 >         status = teste('Material/testes/teste02.jpg')
27
28 test_file_load.py:28:
29 -----
30
31 filepath = 'Material/testes/teste02.jpg'
32
33     def teste(filepath):
34         try:
35 >         if len(imagem)>0:
36 E         NameError: name 'imagem' is not defined
37
38 test_file_load.py:10: NameError
39 ===== 1 failed, 1 error in 0.41 seconds
=====
```

LIMITAÇÃO DA BIBLIOTECA



CONCLUSÃO

A PRINCIPAL CRÍTICA A ESSE MODELO É A INEFICÁCIA EM ALGUM CONTEXTO QUE UTILIZE RESOLUÇÕES MAIORES OU ÂNGULOS DIFERENTES E ATÉ MESMO EM UMA SITUAÇÃO CONDIZENTE A CAPTURA DE FOTOS EM TEMPO REAL, A QUAL AS PESSOAS ESTEJAM SE MOVIMENTANDO LIVREMENTE, ISSO SERIA UMA RESPOSTA A PERGUNTA NORTEADORA. ENTRETANTO, COM EXCEÇÃO DO QUE FOI DITO ACIMA, O BANCO DE IMAGENS FER2013 CUMPRE RAZOAVELMENTE BEM AQUILO QUE PROMETE JUNTO A BIBLIOTECA TENSORFLOW.

REFERÊNCIAS BIBLIOGRÁFICAS

TITAYA, Y; MATTHEW, N. D; TEERADAJ, R. \TEXTBF{MULTIMODAL SENTIMENT ANALYSIS ON VIDEO STREAMS USING LIGHTWEIGHT DEEP NEURAL NETWORKS}. IN: SCITEPRESS P 442-451, DISPONÍVEL EM: [HTTPS://WWW.SCITEPRESS.ORG/PAPERS/2021/103044/103044.PDF](https://www.scitepress.org/papers/2021/103044/103044.pdf), ACESSO EM 19/09/2022.

BALTEZ, R; CESAR, C; CRISTINA, R; GONÇALVES, C; CORREA, G. \TEXTBF{ANÁLISE DE SENTIMENTO PARA REVIEWS APRESENTADOS EM VÍDEOS: MODELO DE REDES NEURAIS REINADO EM BASE DE REVIEWS ESCRITOS}, IN: EAESP, P 2-19, DISPONÍVEL EM: [HTTPS://PESQUISAEASP.FGV.BR/SITES/GVPESQUISA.FGV.BR/FILES/ARQUIVOS/GUS-4-23684-61155-1-PB.PDF](https://pesquisa.eaesp.fgv.br/sites/gvpesquisa.fgv.br/files/arquivos/gus-4-23684-61155-1-pb.pdf), ACESSO EM 20/09/2022.

BARDIN, LAURENCE.\TEXTBF{ ANÁLISE DE CONTEÚDO. TRADUÇÃO DE LUÍS ANTERO RETO E AUGUSTO PINHEIRO}. SÃO PAULO: EDIÇÕES 70 LTDA, 1997, 9 45. DISPONÍVEL EM: [HTTPS://IA802902.US.ARCHIVE.ORG/8/ITEMS/BARDIN-LAURENCE-ANALISE-DE-CONTEUDO/BARDIN-LAURENCE-ANALISE-DE-CONTEUDO.PDF](https://ia802902.us.archive.org/8/items/BARDIN-LAURENCE-ANALISE-DE-CONTEUDO/BARDIN-LAURENCE-ANALISE-DE-CONTEUDO.PDF) ACESSO EM 31/08/2022

IA EXPERT ACADEMY. \TEXTBF{RECONHECIMENTO DE EMOÇÕES COM TENSORFLOW}. DISPONÍVEL EM: [HTTPS://WWW.YOUTUBE.COM/WATCH?V=LRV4MGP_WMK&T=309S](https://www.youtube.com/watch?v=LRV4MGP_WMK&t=309s), ACESSO EM 19/09/2022.

LI, Z; LI, R; JIN, G. \TEXTBF{SENTIMENT ANALYSIS OF DANMAKU VIDEOS BASED ON NAÏVE BAYES AND SENTIMENT DICTIONARY}. IN: IEEE, P 75073 - 75084, DISPONÍVEL EM: [HTTPS://IEEEEXPLORE.IEEE.ORG/DOCUMENT/9060892](https://ieeexplore.ieee.org/document/9060892), ACESSO EM 23/09/2022.

PEREZ, V; MIHALCEA, R; PHILIPPE, L.\TEXTBF{ UTTERANCE-LEVEL MULTIMODAL SENTIMENT ANALYSIS. IN: ACLANTHOLOGY}, P 973-982, DISPONÍVEL EM: [HTTPS://ACLANTHOLOGY.ORG/P13-1096.PDF](https://aclanthology.org/P13-1096.pdf), ACESSO EM 09/08/2022.

WANG, W; WU, J; KAZUAKI, F; WADA, S; KURIHARA, S. VAE-BASED \TEXTBF{ADVERSARIAL MULTIMODAL DOMAIN TRANSFER FOR VIDEO-LEVEL SENTIMENT ANALYSIS}. IN: IEEE, P 51315 - 51324, DISPONÍVEL EM: [HTTPS://IEEEEXPLORE.IEEE.ORG/DOCUMENT/9772490](https://ieeexplore.ieee.org/document/9772490), ACESSO EM 11/10/2022.

AARON, COURVILLE; CARRIER,\TEXTBF{ PIERRE. THE FACIAL EXPRESSION RECOGNITION 2013 (FER-2013)}. DISPONÍVEL EM: [HTTPS://DATAREPOSITORY.WOLFRAMCLOUD.COM/RESOURCES/FER-2013](https://datarepository.wolframcloud.com/resources/fer-2013), ACESSO EM 15/10/2022.

BAI, CHARLES; KHANZADA, AMIL;
CELEPCIKAY, FERHAT. \TEXTBF{RECONHECIMENTO DE EXPRESSÃO FACIAL COM DEEP LEARNING}, P 2, DISPONÍVEL EM: [HTTP://CS230.STANFORD.EDU/PROJECTS_WINTER_2020/REPORTS/32610274.PDF](http://cs230.stanford.edu/projects_winter_2020/reports/32610274.pdf), ACESSO EM 21/10/2022.