

PROTÓTIPO DE UM APLICATIVO PARA DETECTAR EMOÇÕES MEDIANTE O RECONHECIMENTO DE EXPRESSÕES FACIAIS

Autores

José Maurício da Silva¹

Raphael Acácio Jerônimo²

Dawilmar Guimarães de Araújo³

Resumo

Este protótipo tem como objetivo oferecer ajuda aos familiares e cuidadores por meio do reconhecimento da expressão facial usando uma *webcam* com métodos de visão computacional e processamento de imagens para identificar e classificar emoções. O número crescente de pessoas com mais de 60 anos e muitas delas necessitando de cuidados especiais por apresentarem comunicação verbal reduzida justifica este trabalho. Na construção do protótipo, a biblioteca *OpenCV* foi utilizada para capturar as imagens e armazená-las em um diretório e treinar o algoritmo de reconhecimento facial. A linguagem de codificação *Python* foi utilizada com o banco de imagens *fer2013* e a biblioteca *TensorFlow*, para treinar modelos de reconhecimento de emoções, armazenando-a em um diretório para reconhecimento de expressões faciais. Como resultado preliminar, o teste de protótipo do aplicativo utiliza um banco de dados com imagens que simulam a microexpressão facial e imagens de autores que simulam uma expressão de dor, os resultados obtidos obtiveram um índice de confiança de 80%, mas com deficiência de identificação de emoções em um ambiente com pouca luz e mais de 80 cm. O protótipo se mostrou promissor em testes, mas algumas modificações serão necessárias antes da implementação real.

Palavras-chave: Reconhecimento facial. Detecção de faces. Modelagem Computacional. Microexpressões.

Abstract

This prototype aims to offer help to family members and caregivers through facial expression recognition using a webcam with computer vision methods and image processing to identify and classify emotions. The increasing number of people over 60 years of age and many of them needing special care because they have reduced verbal communication justifies this work.. In the construction of the prototype, the OpenCV library was used to capture the images and store them in a directory and train the algorithm for face recognition. The coding language Python was used with the fer2013 image bank and the TensorFlow library, for training emotion recognition models, storing it in a directory for the recognition of facial expressions. As a preliminary result, the application prototype test uses a database with images that simulate facial micro expression and authors' images that simulated an expression of pain, the results obtained a confidence rate of 80%, but with a deficiency to identify to emotions in an environment with low light and more than 80 cm. The prototype showed promise in testing, but some modifications will be needed before actual implementation.

Keywords: Facial recognition. Face detection. Computational Modeling. Micro expressions.

¹ Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas - Faculdade de Tecnologia do Estado de São Paulo. E-mail: contato@fateccruzeiro.edu.br

² Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas - Faculdade de Tecnologia do Estado de São Paulo. E-mail: contato@fateccruzeiro.edu.br

³ Doutorando em Engenharia Mecânica pela Universidade Estadual Paulista Júlio de Mesquita Filho – UNESP e docente da Faculdade de Tecnologia do Estado de São Paulo – FATEC. E-mail: dawilmar.araujo@fatec.sp.gov.br

INTRODUÇÃO

De acordo com a Organização Mundial de Saúde (OMS), o número de pessoas com idade superior a 60 anos chegará a 2 bilhões de pessoas até 2050; isso representará um quinto da população mundial. (JORNAL DA USP, 2018). O Brasil, em 2016, tinha a quinta maior população idosa do mundo, e, em 2030, o número de idosos ultrapassará o total de crianças entre zero e 14 anos. (JORNAL DA USP, 2018, Ministério da Saúde, 2016).

No Brasil existem 5,2 milhões de idosos que necessitam de ajuda para as suas atividades da vida diária. Em pelo menos 80% dos casos, o cuidado é prestado por algum familiar e em 20% este é prestado por uma cuidadora remunerada. Estima-se, portanto, que cerca de 4,2 milhões de familiares cuidam de idosos e 1 milhão de cuidadores sejam contratados ou remunerados (Fiocruz, 2021).

Nos últimos anos pode-se observar um grande aumento do número desses cuidadores, cerca de 547%, passando de 5.263 profissionais em 2007 para 34.051 em 2017 (ESTADÃO, 2018).

Estudos revelam que cuidadores com maior experiência e grau de informação tendem a apresentar desempenho mais adequado quanto ao manejo de sintomas, adotam condutas apropriadas, porém inconsistentes indicando disponibilidade e bom senso no cuidado, porém sem avaliação das consequências do uso de estratégias de manejo, o que implica em muitos casos na necessidade da intervenção de um cuidador profissional (DEGAKI, 2009).

Mesmo os profissionais cuidadores, tendo muita experiência, pode-se ter um alto custo para a própria saúde: Como a depressão, estresse, crises de ansiedade e doenças osteoarticulares por carregar peso esses são alguns dos problemas enfrentados por cuidadores (TENDA, 2016).

Como aliada para maior segurança e confiabilidade podem ser adotadas tecnologias que vêm dar tranquilidade ao cuidador e ao acompanhamento aos idosos. A exemplo, a

Gartner, empresa desenvolvedora de tecnologias relacionadas a introspecção, prevê que até 2022, 10% dos dispositivos pessoais terão recursos de IA emocionais e isso inclui: (GOASDUFF, 2018):

- Jogos de vídeo - Usando visão computacional, o console, vídeo game detecta emoções através de expressões faciais durante o jogo e se adapta a ele.
- Educação - Onde protótipos de *software* de aprendizagem foram desenvolvidos para se adaptar às emoções das crianças. Quando a criança mostra frustração porque uma tarefa é muito difícil ou muito simples, o programa adapta a tarefa para que ela se torne menos

ou mais desafiadora. Outro sistema de aprendizagem ajuda crianças autistas a reconhecer as emoções de outras pessoas.

- Atendimento ao paciente - Um "*bot* de enfermeira" não só lembra pacientes mais velhos em programas médicos de longo prazo para tomar seus medicamentos, mas também conversa com eles todos os dias para monitorar o seu bem-estar geral.
- Segurança do carro - Os vendedores automotivos podem usar a tecnologia de visão computacional para monitorar o estado emocional do motorista. Um estado emocional extremo ou sonolência pode desencadear um alerta para o motorista.
- Serviço público - Surgiram parcerias entre fornecedores de tecnologia de IA e provedores de câmeras de vigilância. Câmeras em locais públicos nos Emirados Árabes Unidos podem detectar expressões faciais das pessoas e, portanto, entender o humor geral da população. Este projeto foi iniciado pelo Ministério da Felicidade do país.

Diante disso, este trabalho apresenta uma proposta alternativa como forma de ajudar aos familiares e cuidadores de pessoas com comunicação verbal reduzida, por meio de um aplicativo que detecta a face e reconhece a emoção nela expressada, utilizando câmeras e aplicando métodos de visão computacional e processamento de imagens para identificação e classificação de emoções.

1. FUNDAMENTAÇÃO TEÓRICA

Conceito de Emoção:

Emoção é uma palavra que deriva do termo em latim *emotione*, cujo significado é mover para fora, promover uma grande agitação que leva algo do interior para a superfície (Equipe IBC, 2020).

Charles Darwin revolucionou a ciência e o nosso modo de vermos o planeta e nossa espécie com a publicação do livro "A Origem das Espécies" (1859). Ele publicou "A expressão das emoções no homem e nos animais" (1872/2000), que é uma das primeiras fontes de informação sobre as emoções dentro da perspectiva evolucionista. Ele descreveu diversas reações emocionais em animais, refletindo sobre sua função biológica e observou as expressões de emoção em seus próprios filhos e em pessoas de outras culturas. *Darwin* defendeu a hipótese da universalidade das expressões faciais ao constatar que certos padrões de expressões emocionais e de reconhecimento destas eram semelhantes entre diferentes culturas.

Durante a década de 1970, o então professor da Universidade da Califórnia *Paul Ekman* realizou estudos transculturais que forneceram evidências em favor da hipótese proposta por *Darwin*. *Ekman* é considerado pela Associação Americana de Psicologia (APA) um dos

psicólogos mais prestigiados e influentes do século XXI. Ele é uma das maiores referências no âmbito da detecção de mentiras e das relações entre as emoções e as expressões faciais.

Paul Ekman constatou em suas pesquisas que a expressão facial de seis emoções básicas (Figura 1) eram as mesmas a toda espécie humana, independente da cultura:

Figura 1. As 6 emoções básicas segundo *Ekman e Friesen*, 1968.



Fonte: Fotos domínio público: <https://clue-lab.com.br/2018/01/04/as-7-emocoes-universais/>. Acesso em 20 fev. 2021.

- Alegria - Uma forte característica da alegria é o famoso pé de galinha, aquelas ruguinhas ao lado dos olhos arregalados. Quando a alegria é falsa, não surgem essas rugas. Sem contar a famosa curva para cima nos lábios. Quanto maior a alegria, maior essa curva.
- Medo - Esta microexpressão é caracterizada por sobrancelhas tensas e olhos muito abertos para ser capaz de visualizar o máximo que pudermos do nosso campo visual, já que estamos percebendo o perigo em algum lugar.
- Nojo - Uma das expressões mais fáceis de identificar por se concentrar basicamente na boca e nariz. Uma das principais características é o nariz enrugado e o lábio superior levantado. Ela também pode ser utilizada em desaprovação a algo ou alguém.
- Raiva - A sobrancelha é um dos pontos principais dessa microexpressão. Normalmente, nós baixamos e juntamos a sobrancelha, fazendo um movimento de franzir a testa. Todo o rosto fica bem tenso. Tensão que também pode ser vista na boca com os lábios separados e dentes apertados.
- Surpresa - Quando demonstramos surpresa, a boca fica solta e aberta como no medo, mas a principal diferença está nas sobrancelhas que ficam arqueadas e com os olhos bem abertos.

- Tristeza - É uma das microexpressões mais difíceis de esconder. As sobrancelhas se juntam no meio, mas de forma sutil e a boca fica um pouco arqueada para baixo.

Mas apenas nos anos 80, *Ekman* adicionou a emoção Desprezo, à lista das emoções básicas e universais. O segredo dessa microexpressão está na parte inferior do rosto. A principal característica é um meio sorriso, com apenas um lado da boca arqueado.

A premissa para que as expressões faciais, de fato, mostram o que sentimos, é a cognição social. Para que uma expressão facial possa ser entendida, o indivíduo receptor deve ter a habilidade de reconhecer a expressão, interpretar o seu significado, entender a sua motivação e responder a ela. Esse processo todo em que acontece essa comunicação não-verbal é a cognição social. (SONSIN, 2017)

Tal meio de comunicação não verbal as expressões e emoções são objeto de estudo não apenas da psicologia, mas também de diversas áreas como filosofia, psiquiatria, biologia e neurociências que passaram a ser mais pesquisadas a partir da segunda metade do século XX. (LIBRALON, 2014)

Atualmente os métodos de reconhecimento da emoção de uma pessoa pelo computador aproximam-se e, em alguns casos, superam o reconhecimento humano. Enquanto o reconhecimento de expressões faciais por humanos é de aproximadamente 87%, alguns algoritmos computacionais, em ambiente controlado, obtêm sucesso entre 74% e 98% (OLIVEIRA E JAQUES, 2013, p.2).

Com estes resultados se torna muito promissor uma aplicação que possa servir de apoio para identificar uma situação de desconforto, ajudando os cuidadores, familiares e assistido.

Nesse sentido, com a finalidade de ajudar tais pessoas, encontrasse em desenvolvimento um aplicativo computacional que identifica determinadas emoções no momento e envia através de aplicativos de mensagens um alarme para os usuários caso as que estejam acompanhando o assistido no momento tenham dificuldades ou não percebam a alteração no comportamento.

1.1 Reconhecimento facial

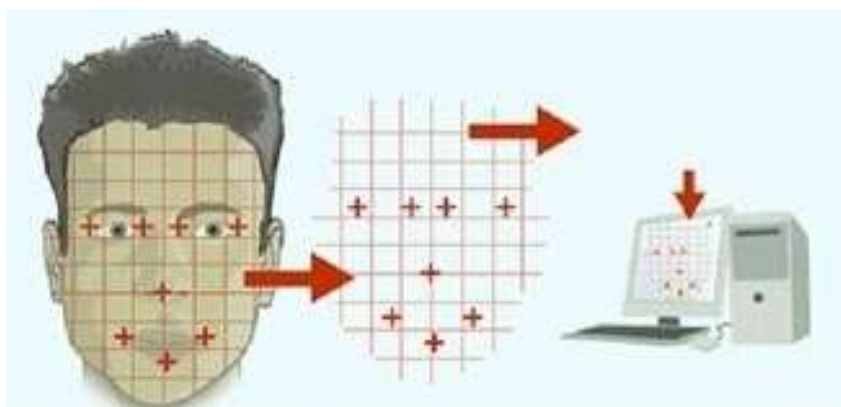
O reconhecimento de uma emoção através da expressão facial em um meio computacional funciona com um sistema que utiliza algoritmos após detectar uma face utilizando-se de uma câmera (*webcam*, celular, de segurança, entre outras).

O rosto humano possui características básicas que não se alteram, que é lida pelos aplicativos como ponto em comum como os dois olhos e a distância entre eles, o nariz e seu

comprimento, a boca, as bochechas e o queixo, limitando assim o formato da face e o espaço ocupado por ela. Este padrão é atingido com um treinamento de um algoritmo em um banco de imagens de expressões faciais de diversas pessoas e vários ângulos.

Os primeiros estudos de um reconhecimento da face humana por um computador se deram no início na década de 1960 pelo matemático e cientista da computação americano *Woodrow Wilson Bledsoe*, ele desenvolveu um sistema que poderia classificar fotos de rostos à mão usando o que é conhecido como um *tablet RAND* (CLISTENIS, 2018). Um *tablet RAND* era um dispositivo que poderia ser usado para inserir coordenadas verticais e horizontais em uma grade usando uma caneta que emite pulsos eletromagnéticos. Este sistema em particular foi usado para registrar manualmente os locais de coordenadas de várias características faciais, como olhos, linha do cabelo, boca e nariz. Um fator comum é encontrar padrões nas faces e dificuldade de reconhecimento de uma face de maneira diferente em que os algoritmos foram treinados, ilustrados na Figura 2.

Figura 2. Esquema do processo de reconhecimento facial.



Fonte: *Face Recognition Solution 2021*

1.2 Inteligência Artificial

Podemos definir inteligência artificial, resumidamente como a capacidade das máquinas aprender, perceber e decidir quais caminhos seguir de forma racional, diante de determinadas situações assim como os seres humanos (COSSETI, 2019)

Tecnologias de reconhecimento facial utilizando inteligência artificial tem sido utilizada no mundo inteiro. Um mapeamento da *Surfshark*, empresa que desenvolve ferramentas de proteção de privacidade na *internet*, revelou que em 2020, 98 países usavam tecnologias de reconhecimento facial em algum tipo de vigilância pública (SCIULO, 2020).

A uma expectativa que esse mercado, estimado em US\$ 3,2 bilhões em 2019, alcance US\$ 7 bilhões em 2024, segundo análise da empresa de pesquisa indiana *MarketsandMarkets*

(MarketsandMarkets, 2020).

2. Materiais e Métodos

No desenvolvimento do *software*, a utilização da linguagem *Python* foi escolhida por conter bibliotecas para análise de dados, inteligência artificial, *Machine Learning* (Python Brasil, 2020).

As *IDE'S* escolhidas para o desenvolvimento do *software* foram o *GOOGLE COLLABORATORY (COLAB)*, que é um serviço de nuvem gratuito hospedado pelo Google para incentivar a pesquisa de Aprendizado de Máquina e Inteligência Artificial, e o *Visual Studio Code* lançado em 2015 pela *Microsoft*, que é um editor de código destinado ao desenvolvimento de aplicações *web*, trata-se de uma ferramenta leve e multiplataforma que está disponível tanto para *Windows*, quanto para *Mac OS* e *Linux* e atende a uma gama enorme de projetos, não apenas *ASP.NET*, como também *Node.js*. Adicionalmente, o editor possui suporte à sintaxe de diversas linguagens como *Python*, *Ruby*, *C + +*.

2.1 Python

É uma linguagem de programação multi-paradigma, pois suporta orientação de objeto, programação imperativa e, em menor escala, programação funcional cuja filosofia enfatiza uma sintaxe favorecendo um código mais legível (Welton Vaz, 2018). Esta linguagem de alto nível que têm uma sintaxe próxima a linguagem humana, que tem uma tipagem dinâmica e de fácil aprendizagem.

Python é uma linguagem muito popular para análise de dados e *data Science* (ZUCHER, 2020). Por possuir bibliotecas como *TensorFlow*, *keras*, *opencv*, *numpy* e *pandas* que são ferramentas para aprendizado de máquina e visão computacional foi essencial na construção deste protótipo.

2.2 Tensor Flow

Tensor Flow é uma biblioteca de código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas se tornou uma das principais ferramentas para *machine learning* e *deep learning*, desenvolvido pela equipe *Google Brain* para uso interno na empresa. Por ser possuir exemplificação detalhada de como treinar uma rede neural e um ambiente completo de treinamento no *Google Collab*, esta biblioteca foi utilizada para facilitar o treinamento e a obtenção dos arquivos necessários para o protótipo (TENSORFLOWG, 2020).

2.3 KERAS

O Keras é uma API (*Application Programming Interface* ou Interface de Programação para Aplicações) de rede neural profunda escrita em *Python*, em execução em cima da plataforma de aprendizado de máquina *TensorFlow* (*keras.io*). A API funcional do Keras é uma maneira de criar modelos mais flexíveis e pode manipular modelos com topologia não linear, camadas compartilhadas e até mesmo várias entradas ou saídas (TENSORFLOW, 2020).

2.4 OpenCV

É uma biblioteca de código aberto para visão computacional, processamento de imagem e aprendizagem de máquina, o projeto foi iniciado na Intel em 1998, com o primeiro lançamento público em 2000, a biblioteca *Open Source Computer Vision Library* (OpenCv). A biblioteca possui mais de 2500 algoritmos otimizados, o que inclui um conjunto abrangente de algoritmos clássicos e de visão computacional e de aprendizado de máquina (OPENCV.ORG). O algoritmo de Viola-Jones, ou *HaarCascade*, ou só *Cascade*, que estão complementando os algoritmos desenvolvidos nesta aplicação é capaz de detectar faces com precisão, alta taxa de acerto, baixa taxa de falsos positivos e baixo custo computacional (SANTANA, GOMES e SANTOS, 2015).

2.5 Selenium

A Empresa *Google* criou uma nova ferramenta de automação de testes chamada *Selenium WebDriver* e *Remote WebDriver* (IBM DEVELOPER).

A ferramenta *Selenium* possui bibliotecas que torna possível a automação em navegadores.

No protótipo sua utilização foi para envio de mensagem automática em caso de detecção de dor no módulo MONITOR DE DOR.

2.6 WhatsApp

O WhatsApp é a rede social de mensagens instantâneas mais popular entre os brasileiros: praticamente todas as pessoas que têm um smartphone também o têm instalado (VOLPATO, 2020).

O WhatsApp surgiu como uma alternativa ao sistema de SMS possibilita o envio e recebimento de diversos arquivos de mídia: textos, fotos, vídeos, documentos e localização,

além de chamadas de voz (WHATTSP, 2021).

Por sua integração com o *Selenium* para automação de mensagem foi incorporado ao protótipo de reconhecimento de emoções.

2.7 Desenvolvimento

O algoritmo para realizar o treinamento do modelo foi feito utilizando *IDE* do *Google Collaboratory (COLAB)* por ser de uso gratuito e de fácil importação de bibliotecas que atendem a proposta da pesquisa.

O *software* foi testado em ambiente domiciliar dos próprios autores utilizando *webcam*, câmera de celular, fotos, vídeos.

Na Figura 3, É apresentado como será a interface gráfica, com a objetivo de ser de fácil interação com usuário possui seis módulos onde o primeiro botão realiza a ação de demonstrar se a face detectada apresenta emoção relacionada a dor ou não, o segundo botão realiza a de mostrar as sete microexpressões faciais e no terceiro botão é possível realizar uma fotografia onde pode se identificar umas das setes microexpressões faciais em todos os módulos logo ao lado do frame é possível visualizar as probabilidades para cada emoção detectada, no quarto botão realiza captura de um vídeo e o grava, mostrando a emoção detectada e o quinto e sexto botão faz o escaneamento de uma foto ou vídeo e mostra a emoção detectada na foto.

Figura 3. Interface gráfica do aplicativo.

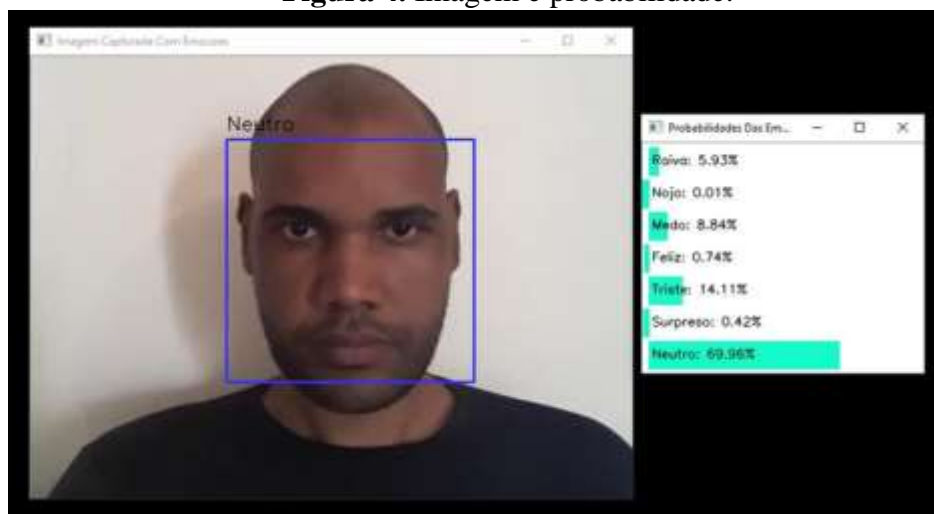


Fonte: Próprios autores.

Na Figura 4, É apresentado como será realizada o reconhecimento das emoções de um determinado usuário, quando o sistema localizar e reconhecer a face, identificar os pontos em

comum e compará-lo ao modelo treinado se há a existência de determinada emoção.

Figura 4. Imagem e probabilidade.



Fonte: Próprios autores.

2.7.1 Treinamento do Modelo para Reconhecer Emoções

Para iniciar um projeto de um sistema para reconhecer emoções faciais, deve-se ter a princípio um banco de imagens contendo apenas a região detectada da face de preferência em escala cinza para um processamento mais rápido e divididas em pastas com os nomes das emoções e as imagens simuladas ou reais, pois a partir delas é possível realizar o treinamento do algoritmo resultando nos modelos de reconhecimento. Utilizamos o banco de imagens de domínio gratuito *FER 2013* que foi utilizado na *KAGGLE* que é uma plataforma para competições de *Machine Learning e Data Science*.

As imagens para o treinamento são fundamentais para um reconhecimento eficiente, as imagens devem ser obtidas de vários ângulos, em ambientes bem iluminados e variação de expressão, feliz, triste, com ou sem óculos (*GRANATYR, 2019*).

2.7.2 Estruturas do Diretório para Treinamento

O treinamento do modelo para reconhecer emoções tem a estrutura de diretórios pré-definidas conforme a sequência abaixo:

- Base com fotos das expressões faciais
- Pré-processamento
- Importando as bibliotecas *TensorFlow/Keras*
- Treinando modelo

2.7.3 Treinamento

O algoritmo para treinamento do modelo inicia no *Google Collaboratory(COLAB)* importando as bibliotecas do *OpenCV* para processos de reconhecimento facial, *Numpy* para realizar os cálculos nos *arrays* multidimensionais, *Pandas* para análise de dados, *Pyplot* para gerar gráficos e o *TensorFlow* para detectar e decifrar padrões e correlações.

Na Figura 5, É apresentado parte do algoritmo para treinamento do modelo onde é feito a importação das bibliotecas.

Figura 5. Importando as bibliotecas.

```
import cv2 # Biblioteca OpemCV
import numpy as np # Biblioteca Numpy para realizar os cálculos nos arrays multidimensionais
import pandas as pd # Biblioteca Pandas para análise de dados
import matplotlib.pyplot as plt # Biblioteca Pyplot para gerar gráficos
import tensorflow # Biblioteca para detectar e decifrar padrões e correlações
```

Fonte: Próprios autores

Os próximos passos do treinamento do modelo é o carregamento das imagens separadas em classes para identificação das faces, o carregamento do *Tenso Flow/keras* para treinamento da rede neural, na sequência passando as classes separadas acima para a arquitetura do modelo(*CNN*) que são Redes Neurais Convolucionais modelo esse desenvolvido por Yann LeCun em 1998(Alves, 2018), passando para *data argumentation*, o objetivo geral do data augmentation é aumentar a generalização do modelo. Na Figura 6, É apresentado parte do algoritmo para treinamento do modelo para reconhecimento de emoções.

Figura 6. Trecho do código de utilização do modelo (*CNN*)

```
# Arquitetura do Modelo (CNN)
num_features = 32
num_classes = 4
width, height = 48, 48
batch_size = 16
epochs = 70

model = Sequential()

model.add(Conv2D(num_features, (3, 3), padding = 'same', kernel_initializer="he_normal",
input_shape = (width, height, 1)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(num_features, (3, 3), padding = "same", kernel_initializer="he_normal",
input_shape = (width, height, 1)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
```

Fonte: Próprios autores

Após finalizar o processo de treinamento do modelo de reconhecimento de emoções temos um arquivo de extensão .h5 que é um arquivo de dados salvo no formato *Hierarchical Data Format (HDF)* que contém conjuntos multidimensionais de dados científicos.

2.7.4 Reconhecimento de Emoções Faciais

Para que seja possível identificar a emoção na face do usuário utilizando o sistema descrito nesse artigo, inicialmente importa-se as bibliotecas *OpenCV* para processos de reconhecimento facial, *Numpy* para realizar os cálculos nos *arrays* multidimensionais, *Time* para obter o tempo do processamento das emoções, o *TensorFlow* para detectar e decifrar padrões e correlações e o *TensorFlow / Keras* que é uma API de alto nível do *TensorFlow* para criar e treinar modelos de aprendizado profundo e carregar o modelo treinado na etapa anterior, agora utilizando a *IDE* da *Microsoft Visual Code*.

Na Figura 7, É apresentado parte do algoritmo para detecção facial e comparação com modelo já treinado, onde é feito a importação das bibliotecas.

Figura 7. Importando as bibliotecas.

```
# IMPORTANDO AS BIBLIOTECAS
import cv2 # Biblioteca OpenCV
import numpy as np # Biblioteca Numpy para realizar os cálculos nos arrays multidimensionais
import time # Biblioteca time para obter o tempo do processamento das emoções
import tensorflow # Biblioteca para detectar e decifrar padrões e correlações

# CARREGANDO O MODELO
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model

model = load_model("Modelos/treinamento_emocoes.h5")
```

Fonte: Próprios autores

O processo de reconhecimento de emoções recebe após o carregamento das bibliotecas do *OpenCV*, *Numpy*, *TensorFlow* e o modelo treinado, a imagem obtida meio de uma câmera ou *webcam*, capturando os frames e redimensionando seu tamanho para que em casos a resolução for de muita alta qualidade o seu processamento não demore.

Na Figura 8, É apresentado parte do algoritmo para detecção facial e comparação com modelo já treinado.

Figura 8. Código para redimensionar a imagem.

TempReal.py > ReconheceTempReal > __init__ > fonte.pequena

```
# ETAPA 4 - REDIMENSIONANDO O TAMANHO (OPCIONAL)
# recomendado quando o tamanho do video é muito grande. Se o video
# tiver resolução muito alta então pode demorar muito o processamento.
redimensionar = True
# define o tamanho da largura (maxima) do video a ser salvo.
# A altura será proporcional e é definida nos calculos abaixo
largura_maxima = 600
if (redimensionar and video.shape[1] > largura_maxima):
    # largura e altura proporcionais (mantendo a proporção do video original)
    # para que a imagem não fique com aparência esticada
    proporcao = video.shape[1] / video.shape[0]
    video_largura = largura_maxima
    video_altura = int(video_largura / proporcao)
else:
    video_largura = video.shape[1]
    video_altura = video.shape[0]
```

Fonte: Próprios autores.

Na sequência das imagens obtidas e redimensionadas, é carregado os algoritmos utilizados para a predição de classes em dados de entrada o *CascadeClassifier* da biblioteca do *OpenCV*, algoritmos esses que são usados na detecção de objetos através de classificadores. Após carregado, o *CascadeClassifier* as imagens obtidas são convertidas para uma escala em cinza para melhor desempenho de limiarização.

Quando o padrão de uma face é encontrado a função *detect Multiscale* do *OpenCV* cria um retângulo com coordenadas (x, y, w, h) ao redor do rosto detectado na imagem, os valores de (x, y, w, h) que determinará o tamanho do retângulo que será criado na face (TABORA, 2019).

Para uma melhor detecção de faces as funções da biblioteca *OpenCv scaleFactor*, *minNeighbors* e *minSize* devem ser ajustadas até encontrar o resultado esperado para a detecção da face.

Na função *scale Factor*, o valor indica quanto o tamanho da imagem é reduzido em cada escala de imagem. Um valor mais baixo usa uma etapa menor para redução de escala. Isso permite que o algoritmo detecte a face. Tem um valor de xy, onde x e y são valores arbitrários que você pode definir.

A função *minNeighbors* especifica quantos "vizinhos" cada retângulo candidato deve ter. Um valor mais alto resulta em menos detecções, mas detecta maior qualidade em uma imagem. Você pode usar um valor de X que especifique um número finito.

A função *min Size* determina o tamanho mínimo do objeto. Por padrão, é (30,30). Quanto menor o rosto na imagem, é melhor ajustar o valor *minSize* mais baixo (TABORA, 2019). O treinamento é realizado com imagens de faces e imagens de outros

objetos assim criando um padrão em imagens que são faces e imagens que não são faces.

Na Figura 9, É apresentado parte do algoritmo para detecção facial e comparação com modelo já treinado, onde é feito a captura facial.

Figura 9. Código para captura e detecção da face.

```

DetectTempReal.py > ReconheceTempoReal > _init_
65 if redimensionar:
66     frame = cv2.resize(frame, (video_largura, video_altura))
67     face_cascade = cv2.CascadeClassifier('Modelos/haarcascade_frontalface_default.xml')
68     cinza = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
69     faces = face_cascade.detectMultiScale(cinza, scaleFactor=1.2,
70     minNeighbors=5, minSize=(30, 30))
71     if len(faces) > 0:
72         for (x, y, w, h) in faces:
73             frame = cv2.rectangle(frame, (x, y), (x + w, y + h + 10), (255, 50, 50),
74             2) # desenho do retangulo da face
75             roi = cinza[y:y + h, x:x + w] # extrai apenas regioao de interesse (ROI) que ond contem o rosto
76             roi = cv2.resize(roi, (
77             48, 48)) # antes de passar para rede neural redimensiona para o tamanho das imagens de treinamento
78             roi = roi.astype("float") / 255.0 # normaliza
79             roi = img_to_array(roi)
80             roi = np.expand_dims(roi, axis=0)
81             # faz a predição - calcula as probabilidades
82             preds = classificador_emocoes.predict(roi)[0]
83             print(preds)
84             emotion_probability = np.max(preds)
85             label = expressoes[preds.argmax()]
86             result = model.predict(roi)[0]
87             # print(result)
88             if result is not None:
89                 resultado = np.argmax(result)
90                 cv2.putText(frame, expressoes[resultado], (x, y - 10), fonte, fonte_media,
91                 (255, 255, 255), 1, cv2.LINE_AA) # escreve a emoção acima
92                 cv2.putText(frame, " Imagem Em Tempo Real Processada Em {:.2f} Segundos".format(time.time() - t),
93                 (20, video_altura - 20), fonte, fonte_pequena, (250, 250, 250), 0, lineType=cv2.LINE_AA)
94             cv2.imshow('P-R-E-F- FaceEmotion', frame)
95

```

Fonte: Próprios autores

2.7.4 Probabilidades

Para calcular as probabilidades referente a lista de emoções foi utilizado a biblioteca *NumPy* do *OpenCv* onde é feito comparação nas listas do modelo de reconhecimento de emoções treinado.

2.7.5 Envio automático de mensagens

O aplicativo de reconhecimento de emoções ao capturar as imagens faciais ira compara las ao modelo treinado e exibirá a emoção que mais se assemelha.

Ao ser monitorado com o módulo de dor, o protótipo de aplicação enviará uma mensagem automática para o número de *WhatsApp* escolhido no momento da conexão.

Ao teclar no Módulo Monitor de Dor do protótipo abre a tela da página *web* do aplicativo de mensagem *WhatsApp*, o usuário deverá se conectar escaneando o *QRCode* que aparecer na tela.

Na Figura 10, apresenta a interface do *WhatsApp Web*.

Figura10. Página web do WhatsApp



Fonte: Próprios autores.

O envio de mensagem acontece quando é detectada uma situação de dor, o navegador *Chrome* foi utilizado na interação com o projeto, para esta interação e necessário o arquivo binário de *ChromeDriver* que deve ser da mesma versão do navegador para que a conexão seja realizada com êxito.

A figura 11, apresenta uma conexão usando *Selenium* e *ChromeDriver*.

Figura11. Código para conexão

```
#conexão para envio de mensagem
message_text='Testando aplicativo. Obrigado pela atenção' # message
no_of_message=1 # numeros de mensagem
#mobile_no_list= [mob] # Colocar os numeros aqui

def element_presence(by,xpath,time):
    element_present = EC.presence_of_element_located((By.XPATH, xpath))
    WebDriverWait(driver, time).until(element_present)

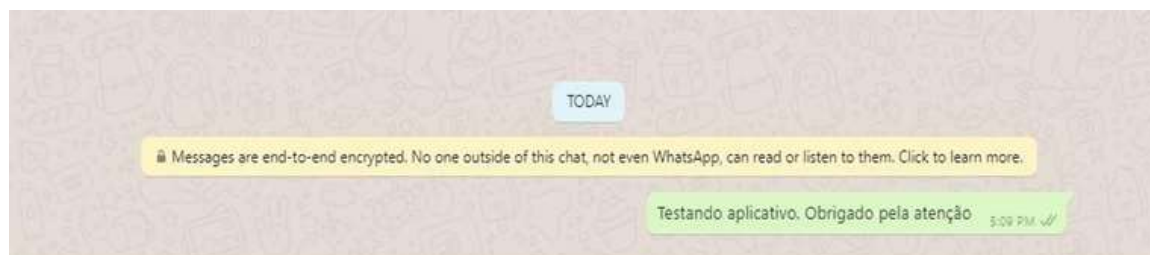
def is_connected():
    try:
        socket.create_connection(("www.google.com", 80))
        return True
    except :
        is_connected()

driver = webdriver.Chrome(executable_path="chromedriver.exe")
driver.get("http://web.whatsapp.com")
sleep(10) # Espera para escanear o QRCode
```

Fonte: Próprios autores.

O monitoramento sendo realizado, quando ocorrer uma situação de dor será enviada a mensagem. A figura 12, representa uma mensagem enviada quando é detectado uma situação de dor pelo protótipo.

Figura12. Exemplo do envio de mensagem Automática

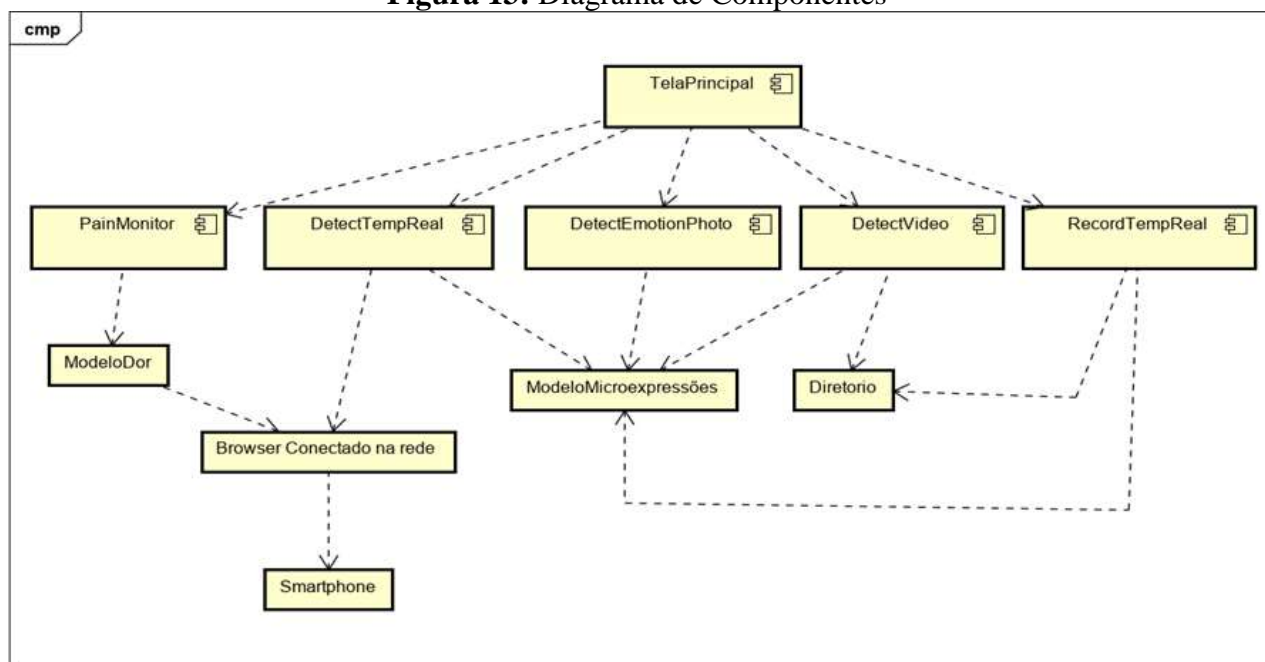


Fonte: Próprios autores.

2.7.6 Diagrama de componentes

O diagrama de componentes mostra o relacionamento entre os módulos do protótipo com capacidade de interação com o restante do sistema e a integração com *web* e outros *Hardware* para o envio de mensagem quando detectada uma situação de dor ou surpresa.

Figura 13: Diagrama de Componentes



Fonte: Próprios autores.

3. RESULTADOS E DISCUSSÕES

Nos testes realizados em ambientes controlados , foram utilizados na execução do programa, 1 *notebook Quad Core* com 4 gigas de memória *RAM* com uma *webcam* de 1,4 megapixel, um *notebook* processador *I5* com 8 gigas de memória *RAM* com uma *webcam* de 7

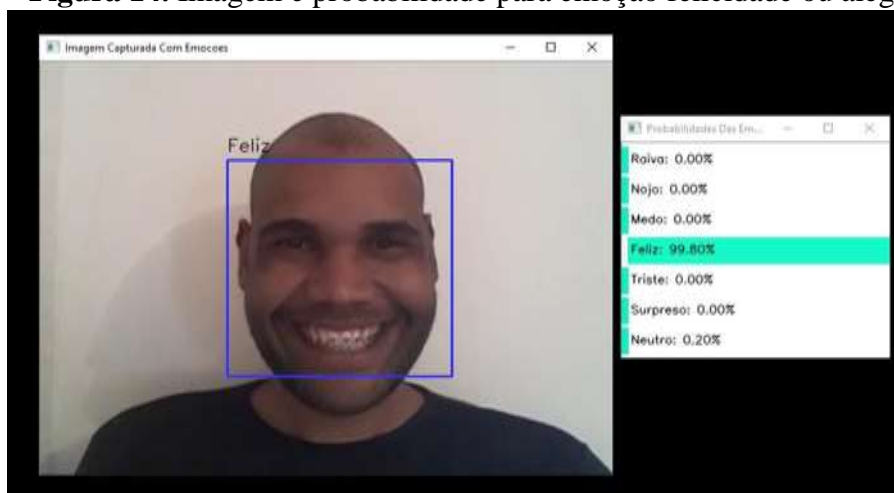
megapixel e foram utilizadas câmeras de 2 celulares com resoluções de 13 e 14 *megapixel* além de imagens de álbuns fotográficos pessoais, os resultados obtidos se mostraram bastante satisfatórios.

Para reconhecer expressões faciais, são necessários dados sobre ações de características faciais, extraídos basicamente de uma imagem (OLIVEIRA E JAQUES, 2013). Depois destas imagens treinadas quando detectar uma face, o protótipo faz uma comparação do treinamento salva em um diretório com a que está detectando no momento e retorna a que mais se assemelha com o treinamento escrevendo acima da imagem a emoção detectada.

Embora a dor seja uma experiência sensorial ou emocional desagradável associada a lesão tecidual, real ou potencial, ou descrita em função dessa lesão (IASP), a ideia de se construir um aplicativo capaz de identificar padrões de desconforto relacionados a dor, é sim de fato viável e de grande utilidade.

Na figura 14, temos a imagem onde é simulada a emoção felicidade ou alegria que segundo *Paul Ekman*(2003) é caracterizada, por aquelas ruguinhas ao lado dos olhos arregalados, quando a alegria é falsa, não surgem essas rugas. Sem contar a famosa curva para cima nos lábios, quanto maior a alegria, maior essa curva. Nessa imagem a probabilidade é de 99.80% para tal emoção em comparação às outras.

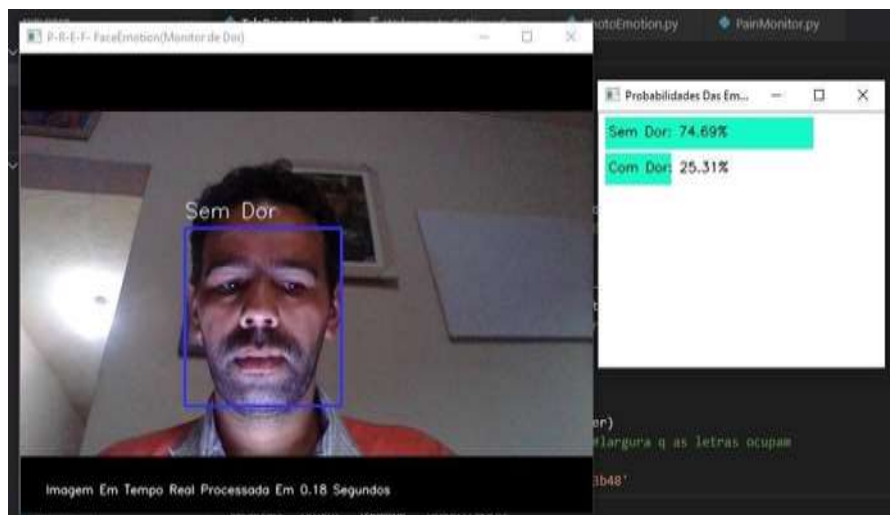
Figura 14. Imagem e probabilidade para emoção felicidade ou alegria.



Fonte: Próprios autores.

Na figura 15, temos a imagem onde não é simulada dor, nessa imagem a probabilidade foi 74.69%.

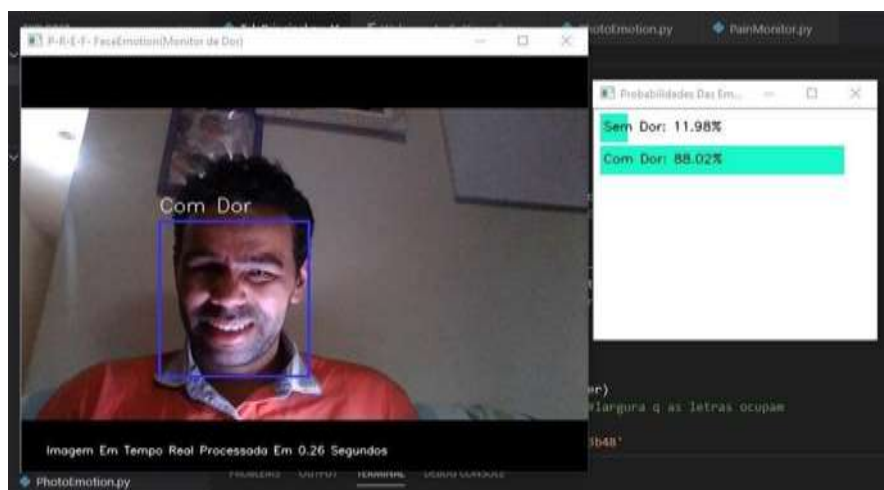
Figura 15. Indica que o usuário não está com dor.



Fonte: Próprios autores.

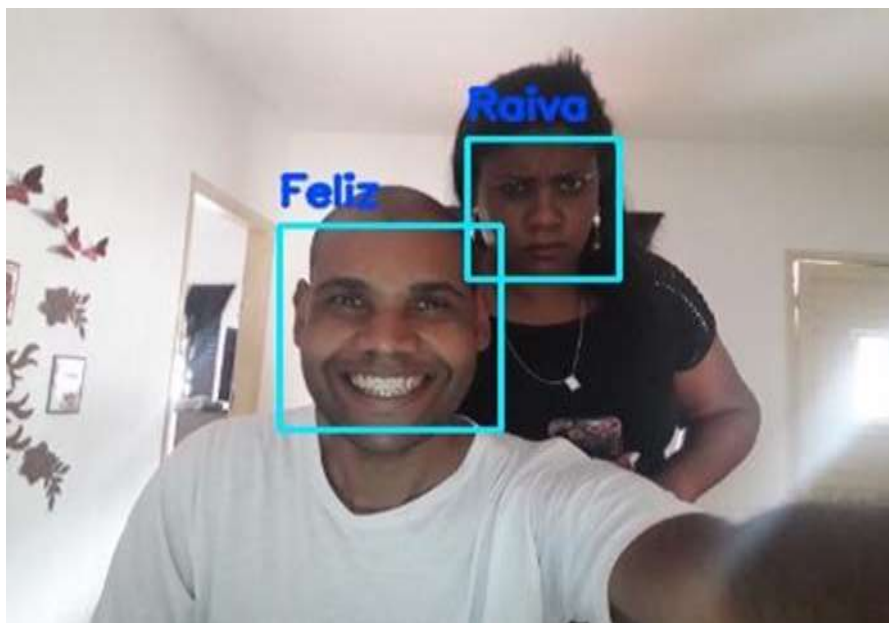
Na figura 16, temos a imagem de como será realizada o reconhecimento dos sintomas de dor com a probabilidade de 88.02% de acerto. Segundo *Toyoshima* (2009), um dos indicativos de dor demonstrado pela face é franzir a testa e fechar os olhos.

Figura 16. Indica que o usuário está com dor.



Fonte: Próprios autores

Figura 17. Simulação das emoções felicidade e raiva – detecção de múltiplas faces.



Fonte: Próprios autores

4. CONSIDERAÇÕES FINAIS

O propósito de desenvolver um aplicativo para reconhecimento de emoções em faces humanas, onde o reconhecimento facial que detecta, captura, processa e reconhece a emoção no momento, tendo como principais bibliotecas o *TensorFlow* e o *OpenCV*, demonstrou-se bastante satisfatório e eficiente para solucionar os problemas proposto na área da visão computacional. Apesar de mostrar um reconhecimento satisfatório de emoções faciais ainda apresenta falhas que torna inviável o projeto, pois as fases de simulação das emoções funcionam muito bem, mas também em alguns momentos não coincidem com a emoção simulada, pois a qualidade das câmeras no momento da captura das imagens e a quantidade de imagens usadas no treinamento do algoritmo é um fator muito importante que deve se levar em consideração.

Por esses e outros motivos o protótipo para detectar emoções faciais, não é indicado para aplicações que monitoram o comportamento de pessoas assistidas, podendo esse projeto ser usado como referência para trabalhos futuros, sendo aperfeiçoado para um resultado mais satisfatório e eficiente.

5. Trabalhos futuros

Apesar de ser um protótipo para fins acadêmicos, e os resultados na identificação das emoções faciais serem bastante satisfatórios pode-se destacar como sugestão para trabalho

futuro:

- Melhor e maior banco de imagens para melhor treinamento dos algoritmos.
- Aprimoramento do aplicativo com foco na plataforma web e mobile.
- Aprimoramento nas notificações através de mensagens instantâneas para celular.
- Melhor desempenho da Inteligência Artificial.

Referências

CLISTENIS, Uberlan. **O que é reconhecimento facial**, 2018. Disponível em: <<http://asppape.org.br/o-que-e-reconhecimento-facial/>>. Acesso em 17 jun.2020.

COELHO, Francisco Leonardo Moro Pereira. **Testes automatizados com Maximo e Selenium WebDriver**, 2014. Disponível em: <<https://developer.ibm.com/br/devpractices/software-development/articles/maximo-selenium-autom-tests/>>. Acesso em mai. 2021.

DESAFIOS de um cuidador informal. **Intellicare**, 2019. Disponível em : <<https://www.intellicare.pt/pt/desafios-um-cuidador-informal/>> Acesso em 10 out. 2020.

EM 2030, Brasil terá a quinta população mais idosa do mundo. **Jornal Da USP**, 2018. Disponível em: <<https://jornal.usp.br/atualidades/em-2030-brasil-tera-a-quinta-populacao-mais-idosa-do-mundo/>>. Acesso em 24 out. 2020.

INTEL. **Opencv**. Disponível em: <<https://opencv.org/intel/>>. Acesso em 17 jun.2020.

GALIMBERTI, Luiz Henrique de Oliveira. **Estudo comparativo de algoritmos de biometria facial disponibilizados pela biblioteca OpenCv para controle de acesso**, 2018. Disponível em: <<https://www.univates.br/bdu/bitstream/10737/2328/1/2018LuizHenriquedeOliveiraGalimberti.pdf>> Acesso em 03 jun.2020.

GOMES, Fábio Rocha. SANTANAS, Luciana Maiara Queiroz de. SANTOS, Thiago S. Reis. **Processo de detecção facial, utilizando viola;Jones**, 2015. Disponível em:<https://www.researchgate.net/publication/318610756_Processo_de_deteccao_Facial_utilizando_ViolaJones>. Acesso em mai. 2020.

GRANATYR, Jones. **Reconhecimento facial com Python e OpenCv**. 2017. Disponível em: <<https://www.udemy.com/course/reconhecimento-facial-com-python-e-opencv/>>. Acesso em 27 fev. 2020.

FABRO, Clara. **O que é API e para que serve? Cinco perguntas e respostas.** TechTudo, 2020. Disponível em: <<https://www.techtudo.com.br/listas/2020/06/o-que-e-api-e-para-que-serve-cinco-perguntas-e-respostas.ghml>>. Acesso em mai. 2021.

GRANATYR, Jones. **Reconhecimento de Emoções com Tensor Flow 2.0 e Python**, 2017. Disponível em: <<https://www.udemy.com/course/reconhecimento-emocoes-tensorflow-20-python/>>. Acesso em 01 abr. 2020.

LIBRALON, Giampaolo Luiz. **Modelagem computacional para reconhecimento de emoções baseada em reconhecimento facial**. 2015. Disponível em: <<https://teses.usp.br/teses/disponiveis/55/55134/tde-10042015-104538/pt-br.php>>. Acesso em 10 de mar. 2020.

Marketsandmarkets. **Facial Recognition Market**, 2020. Disponível em: <www.marketsandmarkets.com/Market-Reports/facial-recognition-market-995.html>. Acesso em: 04 de mai. 2021.

MATTI Pietikäinen, **Scholarpedia is supported by Brain Corporation**, 2010. Disponível em: http://www.scholarpedia.org/article/Local_Binar_Patterns. Acesso em 15 nov. 2020.

OLIVEIRA, Eduardo e AUGUSTIN, Patrícia Jaques. **Classificação de emoções básicas através de imagens capturadas em vídeos de baixa resolução**. Disponível em: <http://seer.upf.br/index.php/rbca/article/viewFile/2807/2374>. Acesso em 20 mar.2020.

POR que o cuidador deve manter a saúde em dia? Tena, 2016. Disponível em: <<https://www.tena.com.br/sobre-a-incontinencia/cuidadores/porque-o-cuidador-deve-manter-a-saude-em-ia#:~:text=Depress%C3%A3o%2C%20estresse%2C%20crises%20de%20ansiedade,Gerontologia%20da%20Universidade%20Federal%20de>>. Acesso em 11 nov. 2020.

PYTHON para quem está começando. **Python**, 2019 Disponível em: <https://python.org.br/introducao/>>. Acesso em 03 jun.2020.

ROSA Keila Maisa Santa. ABREU Renata Silva. BARBOSA Thiago Luís de Andrade. SILVA Carla Silvana de Oliveira. GOMES Ludmila Mourão Xavier. **Desafios e repercussões do trabalho para o cuidador de idosos com doença de Alzheimer**, 2014. Disponível em: <<https://www.efdeportes.com/efd188/o-cuidador-de-idosos-com-doenca-de-alzheimer.htm>> Acesso em 09 out. 2020.

SAMPAIO, Simone. **Conheça as profissões que mais avançaram no Brasil. Entre 2007 e 2017, as ocupações que mais ganharam espaço no mercado de trabalho estão relacionadas a saúde, educação infantil, agroindústria e tecnologia da informação**. Disponível em: <<https://bit.ly/2SvqG0J>>. Acesso em 27 mar.2020.

SALTON, Kelvin. **Updated code**. 2017. Disponível em: <https://updatedcode.wordpress.com/2017/11/26/reconhecimento-facial-como-funciona-o-lbph>. Acesso em 15 nov. 2020.

SILVA, Eliete Da Rosa. **A Importância do Cuidador de Idosos**, 2019. Disponível em <<https://rsaude.com.br/videos/materia/a-importancia-do-cuidador-de-idoso/8008>>. Acesso em 13 nov. 2020.

SOBRE WathsApp. **WathsApp**, 2021. Disponível em: <https://www.whatsapp.com/about/?lang=pt_br>. Acesso em 02 de mai. 2021.

SONSIN, Juliana. **Por que as expressões faciais são importantes, segundo a psicologia?** 2017. Disponível: <https://www.telavita.com.br/blog/expressoes-faciais-e-psicologia/>. Acesso em 15 nov.2020.

TABORA, Vince. **Deteção de rosto usando OpenCV com classificadores Haar Cascade**. Disponível em: <<https://bit.ly/3b2J5bt>>. Acesso em 20 de mai.2020.

TOYOSHIMA, Marcos Tadashi Kakitani. **Dor**, 2009. Disponível em: <https://www.medicinanet.com.br/conteudos/revisoes/1215/dor.htm> >. Acesso em 26 abr. 2021.

VAZ, Welton. **Saiba mais como o python surgiu e qual o seu cenário atual**, 2018. Disponível em: <<https://blog.vulpi.com.br/python-como-surgiu/>> Acesso em 02 de maio. 2021.

VOLPATO, Bruno. **Ranking das redes sociais 2020: as mais usadas no Brasil e no mundo, insights e materiais gratuitos**, 2020. Disponível em: <<https://resultadosdigitais.com.br/blog/redes-sociais-mais-usadas-no-brasil/>>. Acesso em 02 de mai. 2021.

WINDOWS. **Unofficial windows binaries for python extension packages**. Disponível em: <<https://www.lfd.uci.edu/~gohlke/pythonlibs/>>. Acesso em 10 de abr. 2020.

ZUCHER, Vitor. **Python: O guia completo para aprender a programar nessa linguagem do zero**, 2020. Disponível em: <<https://www.lewagon.com/pt-BR/blog/guia-aprender-programarpython#:~:text=Python%3A%20O%20guia%20completo%20para,linguagem%20do%20zero%20%7C%20Le%20Wagon>>. Acesso em 05 de jan. 2021.

ICICT/FIOCRUZ. **Nota técnica sobre vacinação de idosos e cuidadores**, 2021. Disponível em: <[//portal.fiocruz.br/noticia/nota-tecnica-sobre-vacinacao-de-idosos-e-cuidadores](https://portal.fiocruz.br/noticia/nota-tecnica-sobre-vacinacao-de-idosos-e-cuidadores)>. Acesso em 16 de jun. 2021.