

# **Relatório de Trabalho Visão Computacional**

## **Grupo:**

Nome: Gabriela Prestes Farias | Matrícula: 03058613

Nome: George Moraes de Sousa | Matrícula: 03113538

Nome: José Ribamar Queiroz da Silva Neto | Matrícula: 03118421

Nome: Sthefanye Guimarães Oliveira | Matrícula: 03116527

**Turma: UNN0790107NNA**

**Manaus – AM**

# Sumário

Sobre o projeto	3
Preparativos para a execução do projeto	4
Executando o projeto	5
Referências	7

# Sobre o projeto

O trabalho trata-se de um projeto que realiza o reconhecimento facial do usuário e também detecta se ele está ou não sem máscara.

Desenvolvido na linguagem python, o código utiliza da biblioteca OpenCV para o processamento das imagens e scikit-learn para o treinamento do modelo que classifica um rosto que porta ou não uma máscara.

OpenCV é uma biblioteca de programação, de código aberto, é usada para diversos tipos de análise em imagens e vídeos, como detecção, tracking e reconhecimento facial, edição de fotos e vídeos, detecção e análise de textos, etc.

Para a detecção das faces, utiliza-se o Cascade Classifier, já incluso dentro da biblioteca do OpenCV. De modo geral, esse método de treinamento utiliza de um arquivo .xml, que também já é incluso no pacote, para treinar um modelo que reconhece rostos de forma genérica, utilizando o método Viola-Jones e AdaBoost para o melhoramento do desempenho.

O algoritmo de Machine Learning escolhido para classificação foi o K-Nearest Neighbor, um dos algoritmos de aprendizado de máquina mais famosos em sua caixa de ferramentas de aprendizado de máquina. Com ele, realizamos o treinamento da IA, para ficar mais preciso o reconhecimento facial do usuário.

# Preparativos para a execução do projeto

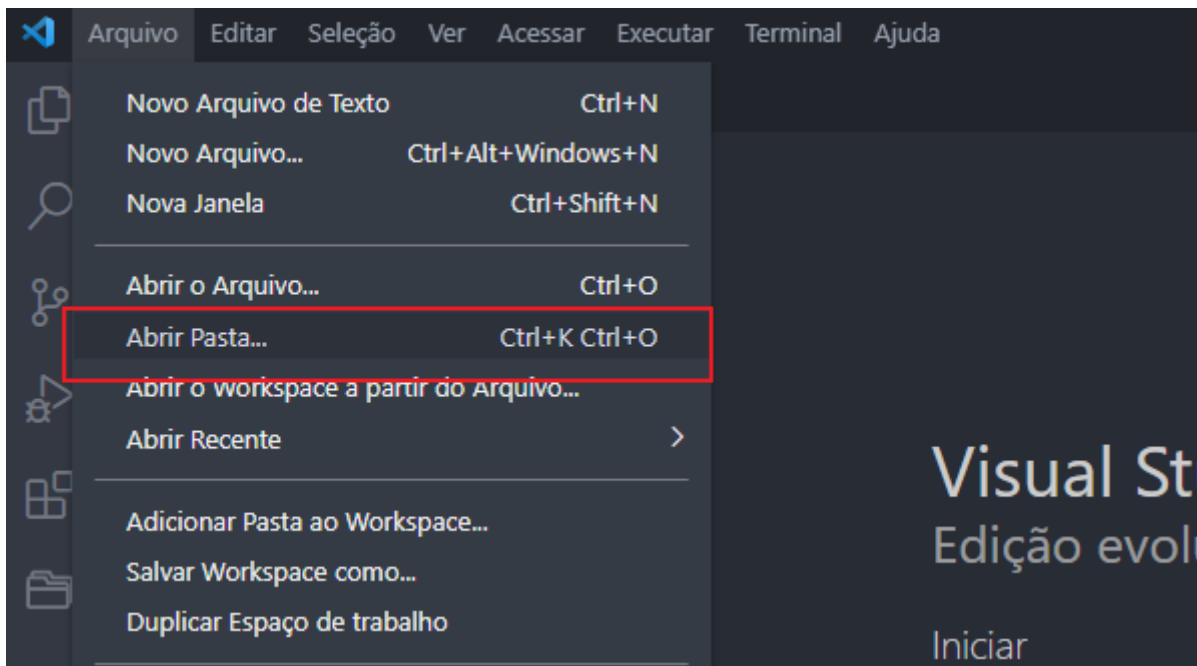
Primeiramente, precisa-se ter instalado no computador, os seguintes softwares:

- Git Bash: [link para download do Git](#)
- VS Code: [link para download do VS Code](#)
- Python: [link para download do Python](#)

Após a instalação do Git, no terminal do git bash ou shell será necessário clonar o repositório, rodando o seguinte comando: **git clone** <https://github.com/Sthefanye/IA.git>

```
➜ git clone https://github.com/Sthefanye/IA.git
Cloning into 'IA' ...
remote: Enumerating objects: 881, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 881 (delta 1), reused 6 (delta 0), pack-reused 872
Receiving objects: 100% (881/881), 17.69 MiB | 10.00 MiB/s, done.
Resolving deltas: 100% (34/34), done.
```

Depois de clonar o repositório, deve-se abrir a IDE de desenvolvimento, e abrir a pasta em que o repositório foi clonado.



Ainda no terminal, utilize o PIP ( gerenciador de pacotes do python) para instalar todos os requisitos.

- `pip install -r requirements.txt`

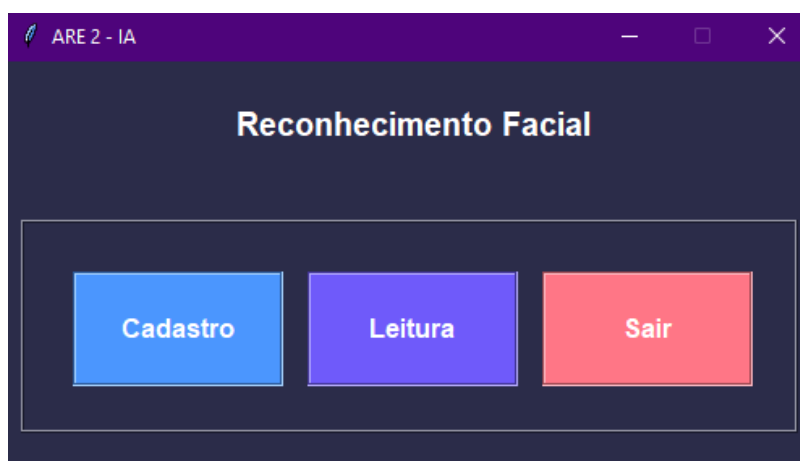
```
➜ pip install -r requirements.txt
```

# Executando o projeto

Após a instalação dos requisitos, executar o projeto, rode o comando no seu terminal.

```
-> py reconhecimentoFacial.py
```

Irá executar e aparecer o menu de opções para o usuário.

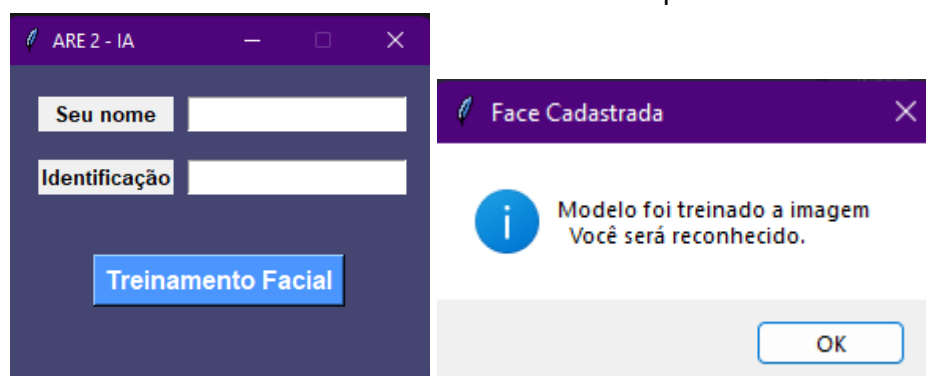


- **Cadastro:** Etapa de cadastrar sua face para armazenar no projeto.

- **Leitura:** Onde abrirá a webcam do dispositivo para realizar o reconhecimento facial.

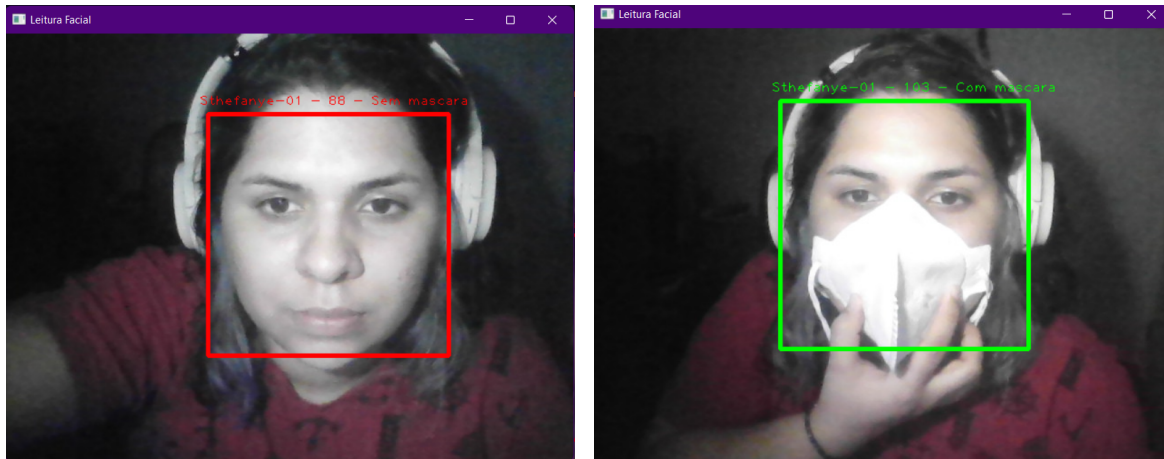
- **Sair:** Sair do programa.

- **Cadastro:** Ao clicar na opção de cadastro, irá aparecer outro menu, para você inserir o nome e ID do usuário. Terá que ser inserido somente o primeiro nome, sem espaços e caracteres especiais. Após inserir os dados, o programa irá abrir a webcam e realizar o cadastro do rosto do usuário. Depois lhe trará um feedback.



- **Leitura Facial:** Depois de realizar o cadastro, clicando em leitura, após alguns segundos, a webcam abrirá e o reconhecimento facial será feito.  
*Obs.: Para fechar a webcam é só clicar na tecla "Q".*

Exemplo de como a leitura facial ocorre:



O usuário precisa estar em um local iluminado, para que o reconhecimento facial seja mais preciso.

- **Sair:** após clicar na tecla 'Q' para fechar a webcam, o usuário pode clicar no botão de sair para fechar a janela.

# Referências

ADRIANO, Ronaldo. OpenCV: Uma breve introdução à visão computacional com python. *In*: OpenCV: Uma breve introdução à visão computacional com python. [S. l.]: Cedro Technologies, 7 out. 2021. Disponível em: <https://blog.cedrotech.com/opencv-uma-breve-introducao-visao-computacional-com-python>. Acesso em: 26 maio 2022.

A ENCICLOPÉDIA LIVRE, WIKIPEDIA. Scikit-learn. *In*: Scikit-learn. [S. l.], 16 jun. 2021. Disponível em: <https://pt.wikipedia.org/wiki/Scikit-learn>. Acesso em: 26 maio 2022.

ROBBISON, Scott. K-Nearest Neighbors: Algorithm in Python and Scikit-Learn. *In*: K-Nearest Neighbors: Algorithm in Python and Scikit-Learn. [S. l.], 21 nov. 2021. Disponível em: <https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/>. Acesso em: 26 maio 2022.

OPEN, CV. OpenCV: Documentation. *In*: Open CV. [S. l.], 27 jan. 2022. Disponível em: <https://docs.opencv.org/4.x/index.html>. Acesso em: 27 maio 2022.