



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Reconhecimento Facial com Raspberry Pi 3

Wglastonio L Sousa, Marcelo T Azevedo

Universidade Presbiteriana Mackenzie (UPM)

Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brasil

Abstract:

This paper presents a project developed for the discipline “Objetos Inteligentes Conectados” which is based on Raspberry Pi module. It is a small system for face recognition based on Raspberry Pi 3, using the OpenCV library which is responsible for the real-time image processing using machine learning algorithms. For records, this project was built just for learning objectives.

Resumo:

Este artigo apresenta um trabalho prático desenvolvido para a disciplina Objetos Inteligentes Conectados baseado no módulo Raspberry Pi 3. Trata-se de sistema de reconhecimento facial baseado no Raspberry Pi 3, utilizando basicamente a biblioteca OpenCV responsável pelo processamento de imagem em tempo real através de algoritmos de “machine learning”. É importante deixar registrado que esse projeto foi construído apenas para fins de aprendizado.

1. Introdução

Conforme visto na disciplina de Objetos Inteligentes Conectados, a IoT – Internet das Coisas, está evoluindo como uma estrutura distribuída, envolvendo diferentes fornecedores desde o nível de componentes básicos, como sensores, passando pelos diversos dispositivos até as plataformas de monitoramento, controle e processamento de dados. Com os esforços unificados de fabricantes, as padronizações têm evoluído de modo haver maior e melhor integração entre os

componentes e soluções. O avanço do 5G, sem dúvida, será essencial na expansão de aplicações IoT nos diversos segmentos de negócios.

O projeto apresentado é uma solução de reconhecimento facial utilizado para identificação de pessoas. Um assunto em alta, visto a busca contínua por processos mais eficientes e precisos de autenticação humana, para acesso aos diversos recursos e serviços disponibilizados via meio eletrônico, seja para acesso a um edifício até acesso a conta bancária. Dentre os diversos métodos de autenticação, o reconhecimento facial ficou mais popularizado para os usuários de aparelhos celulares, onde é utilizado para desbloquear o aparelho além de autenticar o usuário para utilização de aplicativos mais específicos. Também é utilizado por usuários de computadores, porém em menor escala. Considerando as soluções de IoT, há sistemas de autenticação facial para acesso a residências ou áreas restritas em uma empresa, por exemplo. Além disso, com a pandemia recente do coronavírus, esse método de autenticação apresentou crescimento pois pode ser realizado sem a necessidade de o usuário tocar em nada.

Apesar das vantagens apresentadas pela tecnologia, ela ainda não é unanimidade dentre os diversos desenvolvedores. Um exemplo é a IBM, que decidiu encerrar sua área de pesquisa em reconhecimento facial por não concordar com o uso da tecnologia para fins de vigilância em massa e violação dos direitos humanos.

Esse projeto é apenas para fins didáticos para demonstrar a capacidade do módulo Raspberry Pi como elemento IoT e adota como base o artigo *How to Train your Raspberry Pi for Facial Recognition* apresentado no Tom's Hardware, uma publicação especializada em temas de tecnologia desde 1996. O projeto foi desenvolvido sobre a plataforma Raspberry Pi 3. Utiliza a biblioteca OpenCV, de código aberto, para o processamento de imagens em tempo real através de técnicas de “machine learning”. Também utiliza o pacote “face_recognition” responsável pela identificação das fronteiras através de uma marcação em torno da face e por realizar as devidas comparações com o “dataset” existente. Esse “dataset” é o repositório com as imagens pessoais coletadas durante a fase de treinamento do modelo.

2. Materiais e Métodos

Como já mencionado, o projeto a ser desenvolvido terá como base a plataforma Raspberry Pi, conforme apresentado na Figura 1.



Figura 1 – Raspberry Pi 3 e conexões

Em resumo, a primeira etapa para o desenvolvimento do projeto de reconhecimento facial foi reunir os materiais a serem utilizados, os quais estão descritos a seguir:

- Raspberry Pi (versão 4)
- SD Card 32GB
- Monitor
- Teclado
- Mouse
- Câmera
- Fonte de alimentação

Para o funcionamento do sistema de reconhecimento facial, serão utilizadas as bibliotecas **OpenCV**, que é uma biblioteca *open source* utilizada no processamento de imagens e vídeo em tempo real, baseado nas características e aprendizado de máquina, a biblioteca *face_recognition*, responsável pela criação das fronteiras em torno da face, processamento de características e comparação com o dataset, e o pacote *imutils*, integrado ao OpenCV e implementa funções responsáveis pelo processamento de imagens como, translação, rotação, redimensionamento,

estruturação, apresentação, detecção de bordas e contornos, dentre outras. Esse conjunto de bibliotecas será utilizado para executar o treinamento do modelo a partir do *dataset* disponibilizado. O script “train_model.py” é responsável por analisar as imagens dentro do *dataset* e criar um mapeamento entre nomes e respectivas imagens de face dentro do arquivo “encodings.pickle”. Após finalizar o treinamento do modelo, utilizaremos o script “facial_req.py” para detectar e identificar faces.

No diagrama simplificado, apresentado na Figura 2, podemos observar mais claramente as conexões dos componentes utilizados no projeto.

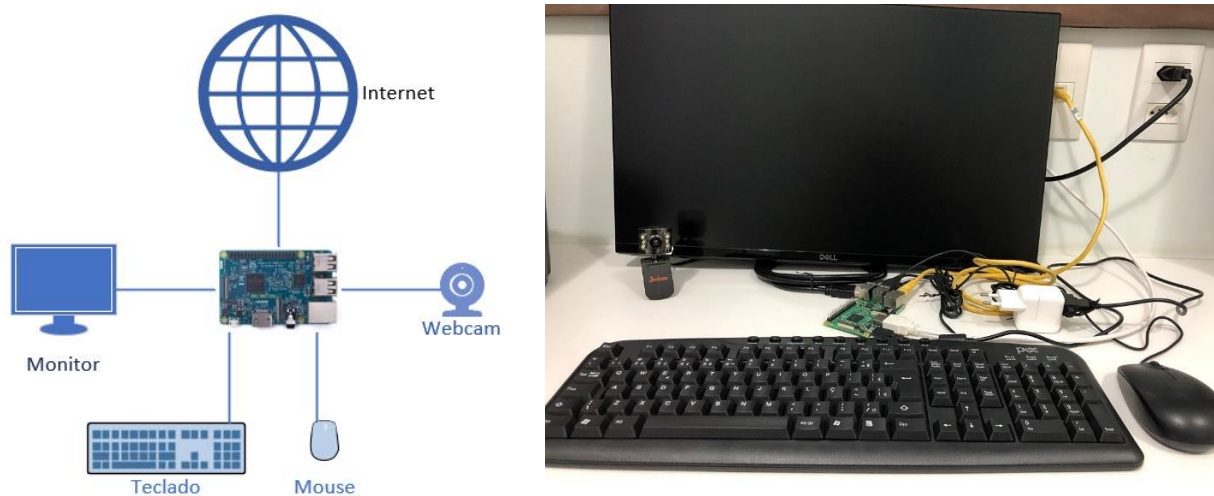


Figura 2 – Raspberry Pi 3 e conexões

A seguir, descrevemos as principais etapas para a construção do protótipo de reconhecimento de faces baseado no Raspberry Pi.

- Instalação:
 - Conexão da câmera via interface USB.
 - Preparar um cartão SD com o sistema operacional do Raspberry Pi e ligar o módulo.
 - Após iniciar o módulo, seguiremos com a instalação das bibliotecas OpenCV, face_recognition, e imutils. Após isso, serão atualizadas as configurações.
- Treinamento do modelo:

- Para o treinamento do modelo é utilizado o *dataset* que acompanha o pacote de software principal.
- Além disso, são utilizadas imagens próprias para complementar o treinamento, as quais são referenciadas dentro da mesma pasta de arquivos. Para isso, devemos obter algumas fotos.
- Como se trata de um modelo de dados, quanto mais amostras fornecermos, mais preciso será o reconhecimento.
- Após a captura das imagens, deveremos executar o treinamento especificamente.

Com o material disponível, iniciamos a segunda etapa do projeto, que foi a instalação do sistema operacional do Raspberry Pi no SD Card. Para isso, baixamos uma ferramenta disponível no site do projeto Raspberry a qual realiza a instalação da imagem do sistema operacional no SD Card. O sistema operacional do Raspberry Pi é baseado na distribuição Debian do Linux e essa etapa foi feita a partir de um computador local.

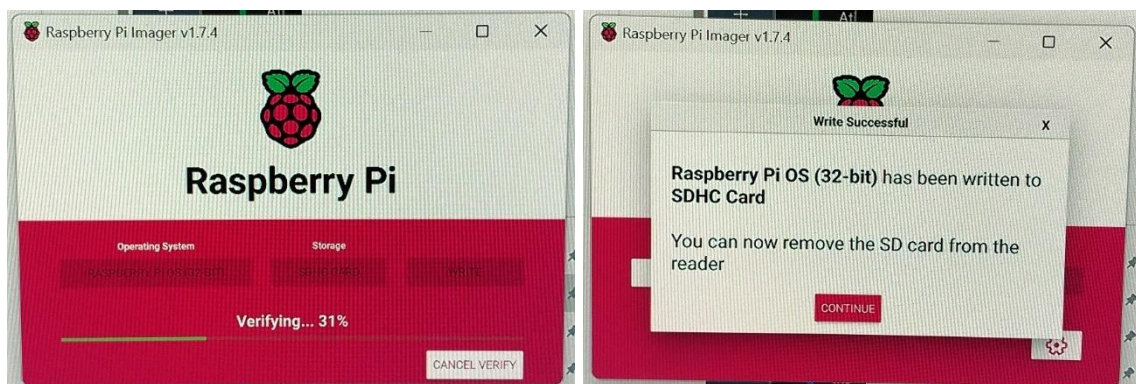


Figura 3 – Preparação do sistema operacional

Concluída a instalação do sistema operacional, seguimos para a terceira etapa onde foi realizada a instalação da biblioteca OpenCV. Essa foi uma das fases mais demoradas pois exige a correta instalação de pacotes complementares para o correto funcionamento da biblioteca OpenCV. Após isso, realizamos o clone do projeto OpenCV a partir do Github, feita a compilação do projeto dentro do Raspberry e, em seguida a instalação. A seguir é descrito o passo a passo:

PACOTES COMPLEMENTARES:

1) Atualização de pacotes:

```
sudo apt update  
sudo apt upgrade
```

2) Pacotes usados para compilar o OpenCV:

```
sudo apt install cmake build-essential pkg-config git
```

3) Pacotes para suporte a diferentes tipos de imagens:

```
sudo apt install libjpeg-dev libtiff-dev libjasper-dev  
libpng-dev libwebp-dev libopenexr-dev  
sudo apt install libavcodec-dev libavformat-dev  
libswscale-dev libv4l-dev libxvidcore-dev libx264-dev  
libdc1394-22-dev libgstreamer-plugins-basel.0-dev  
libgstreamer1.0-dev
```

4) Pacotes de interface:

```
sudo apt install libgtk-3-dev libqt5gui5 libqt5webkit5  
libqt5test5 python3-pyqt5
```

5) Pacotes para otimização de velocidade do OpenCV:

```
sudo apt install libatlas-base-dev liblapacke-dev gfortran
```

6) Pacotes do Hierarchical Data Format (HDF5) usados pelo OpenCV para gerenciamento de dados:

```
sudo apt install libhdf5-dev libhdf5-103
```

7) Pacotes para compilar o OpenCV com suporte a Python:

```
sudo apt install python3-dev python3-pip python3-numpy
```

PREPARAÇÃO PARA COMPILAR O OPENCV:

1) Aumentar o espaço de swap:

```
sudo nano /etc/dphys-swapfile  
> Procurar: CONF_SWAPSIZE=100  
> Substituir por: CONF_SWAPSIZE=2048  
> Salvar e sair
```

2) Reiniciar o serviço:

```
sudo systemctl restart dphys-swapfile
```

3) Clonar os repositórios do OpenCV:

```
git clone https://github.com/opencv/opencv.git  
git clone https://github.com/opencv/opencv_contrib.git
```

COMPILAÇÃO E INSTALAÇÃO DO OPENCV

1) Criar uma build a partir do projeto clonado:

```
mkdir ~/opencv/build  
cd ~/opencv/build
```

2) Preparar o OpenCV para compilação:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \  
-D CMAKE_INSTALL_PREFIX=/usr/local \  
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \  
\  
-D ENABLE_NEON=ON \  
-D ENABLE_VFPV3=ON \  
-D BUILD_TESTS=OFF \  
-D INSTALL_PYTHON_EXAMPLES=OFF \  
-D OPENCV_ENABLE_NONFREE=ON \  
-D CMAKE_SHARED_LINKER_FLAGS=-latomic \  
-D OPENCV_PYTHON_INSTALL_PATH=lib/python3.9/dist-  
packages \  
-D BUILD_EXAMPLES=OFF ..
```

3) Compilar o OpenCV (esse processo demora em torno de 4 horas para ser concluído):

```
make -j$(nproc)
```

4) Instalar o OpenCV:

```
sudo make install
```

5) Atualizar as bibliotecas no sistema operacional

```
sudo ldconfig
```



Figura 4 – Processo de instalação do OpenCV

VOLTAR AS CONFIGURAÇÕES DE SWAP:

1) Aumentar o espaço de swap:

```
sudo nano /etc/dphys-swapfile  
> Procurar: CONF_SWAPSIZE=2048  
> Substituir por: CONF_SWAPSIZE=100  
> Salvar e sair
```

2) Reiniciar o serviço:

```
sudo systemctl restart dphys-swapfile
```

INSTALAR BIBLIOTECAS COMPLEMENTARES

1) face_recognition

```
pip install face-recognition
```

2) imutils

```
pip install imutils
```

3. Resultados

Uma vez que a instalação foi concluída, executamos a etapa de treinamento do modelo. Para isso, devemos baixar o pacote *facial_recognition* a partir do github:

```
git clone https://github.com/carolinedunn/facial_recognition
```

Após isso, criamos a pasta correspondente ao nome da pessoa para armazenamento das respectivas imagens que serão utilizadas no treinamento do modelo:

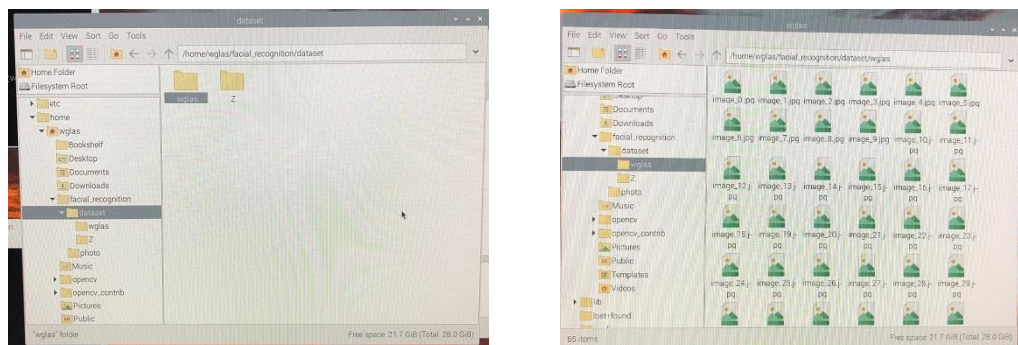


Figura 5 – Criação do dataset

Através do script “headshots.py” realizamos a captura das amostras de imagens da face. Após a captura das imagens para composição do *dataset*, realizamos o treinamento do modelo através do script “train_model.py”. Esse script Python, busca as imagens depositadas na respectiva pasta e a utiliza para criação do modelo de reconhecimento. Depois disso, utilizamos efetivamente o script de reconhecimento “facial_req.py” para identificar uma face, conforme.

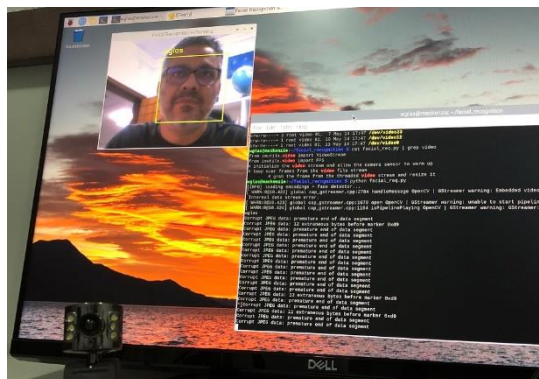


Figura 6 – Reconhecimento de face

Uma vez que o sistema apresentou resultado conforme esperado, ou seja, realizou o reconhecimento da primeira face corretamente, foi adicionada uma segunda face ao *dataset*. Assim, seguimos o mesmo processo, criação da pasta para depósito das novas imagens, adicionados *screenshots* da segunda face, realizado o treinamento do *dataset* e, em seguida, realizado o teste de reconhecimento com sucesso.



Figura 7 – Reconhecimento de outra face

Adicionalmente aos resultados apresentados nesse relatório, foi criado um vídeo mostrando o projeto por etapas. O vídeo está disponível no seguinte endereço abaixo e ficará disponível até o dia 15 de julho de 2023: <https://1drv.ms/v/s!Aj4lt4yd8DKMjuZIlkRccxfyMtfTlw?e=o2DchE>

E os principais componentes do projeto estão disponíveis no github:

https://github.com/wglastonio/face_recognition

s

4. Conclusões

Esse foi um projeto bastante desafiador, visto os diversos módulos de software que fazem parte da solução final. Além disso, também temos que considerar que o projeto foi instalado em uma plataforma com baixo poder computacional, o que traz alguns desafios adicionais durante o processo de instalação. Outro desafio na construção desse projeto, foi em relação a algumas bibliotecas desatualizadas, o que exigiu algumas etapas adicionais de *troubleshooting* e pesquisa para buscar as respectivas versões mais novas e também compatíveis com o restante dos pacotes. Por último, o script utilizado efetivamente para o reconhecimento de face apontava para um device câmera que não estava mapeado, assim foi necessário identifica-lo e efetuar a correção.

Apesar dos fatores limitantes, foi possível observar a capacidade do módulo Raspberry Pi em prover uma solução tecnológica altamente utilizada atualmente. Ou seja, a possibilidade de podermos ter um sistema de reconhecimento facial em uma plataforma tão pequena, permite inúmeras possibilidades de aplicações. Pensando em controle de acesso, por exemplo, há aplicações residenciais, em condomínios, em veículos de um modo geral e assim por diante. Do mesmo modo, também poderia ser utilizada em sistemas de autenticação, por exemplo, para uso de um caixa eletrônico, pagamento de contas, autorizações em geral, dentre outras.

Como ponto de melhoria do projeto, poderíamos propor a implementação da funcionalidade de notificação por email. Trata-se do envio de email sempre que uma pessoa for identificada. Um exemplo de aplicação nesse caso, supondo que essa solução seja utilizada para reconhecimento na entrada de uma residência, seria notificar via email que determinada pessoa está esperando na porta.

5. Referências

DUNN, Caroline. How to Train your Raspberry Pi for Facial Recognition. *Tom's Hardware*, 17 set. 2022. Disponível em: <https://www.tomshardware.com/how-to/raspberry-pi-facial-recognition>. Acesso em: 15 fev. 2023.

GEITGEY, Adam. Face Recognition. *Python Institute*, 20 fev. 2020. Disponível em: <https://pypi.org/project/face-recognition/>. Acesso em 18 fev. 2023.

Open CV – Open Source Computer Vision Library. Copyright 2023. Disponível em: <https://opencv.org/about/>. Acesso em 18 fev. 2023.

Forbes Insider. IBM encerra área de reconhecimento facial e pede reforma da polícia. 9 jun. 2020. Disponível em: <https://forbes.com.br/forbes-tech/2020/06/ibm-encerra-area-de-reconhecimento-facial-e-pede-reforma-da-policia/>. Acesso em: 19 fev. 2023.

PILTCH, Avram. How to Set Up a Raspberry Pi for the First Time. *Tom's Hardware*, 21 set. 2022. Disponível em: <https://www.tomshardware.com/how-to/set-up-raspberry-pi>. Acesso em: 15 abr. 2023.

Raspberry Pi OS. Raspberry Pi Foundation. Disponível em: <https://www.raspberrypi.com/software/>. Acesso em 21 abr. 2023.

SATO, Mia. The pandemic is testing the limits of face recognition. *Mit Technology Review*, 28 set. 2021. Disponível em: <https://www.technologyreview.com/2021/09/28/1036279/pandemic-unemployment-government-face-recognition/>. Acesso em: 19 fev. 2023.

YOUNG, Emmet. Installing OpenCV on the Raspberry Pi. *Pi My Life Up*, 20 set. 2022.
Disponível em: <https://pimylifeup.com/raspberry-pi-opencv/>. Acesso em: 15 abr. 2023.