

Controle de acesso sem fio com reconhecimento facial

Giovanni Bruno Molitor Schiffini¹; Gustavo Mansur de Oliveira² &

Henrique Soares Lima Costa³

Orientador: Prof. Mário C. F. Garrote.

Co-orientador: Prof. Renato Camargo Giacomini.

TG-01-2018-EL

^{1,2,3} Graduandos do Curso de Engenharia Eletrônica, da USJT, turma de 2018.

e-mail¹: giovanni_bruno_20@hotmail.com; e-mail²: gustavo.mansurdeoliveira@hotmail.com;

e-mail³: ricksoares01@gmail.com.

Resumo – A segurança predial e o controle de acesso são preocupações cada vez maiores nos edifícios modernos. Nos últimos anos houve grandes avanços nesta área, como a implementação de controle de acesso por cartão RFID, bluetooth, NFC, biometria e até mesmo validação por aplicativo para smartphone. Ainda assim, a maioria dos edifícios continua utilizando sistemas de controle de acesso que requerem a inserção manual de novos usuários, gerando grandes filas nas recepções e halls de entrada. Partindo dessa necessidade, foi desenvolvido um sistema de controle de acesso por reconhecimento facial que possibilita que o próprio usuário se cadastre e que as portas sejam liberadas via wi-fi. Este artigo tem como objetivo apresentar as características e a metodologia utilizada para a concepção deste projeto.

Abstract – Property security and access control are growing concerns in modern buildings. In recent years there have been major advances in this area, such as the implementation of access control by RFID cards, bluetooth, NFC, biometry and even app validation for smartphones. Still, most buildings continue to use access control systems that require the manual insertion of new users, generating large queues at receptions and entrance halls. Therefore, a facial recognition access control system was developed allowing the user to register himself and the release of the doors via wi-fi. This paper aims to present the characteristics and the methodology used to design this project.

Palavras-chave: acesso, cadastro, reconhecimento, wi-fi.

Introdução

Atualmente, a automação não se restringe apenas a área industrial, como ocorria no século passado. Os edifícios comerciais, como condomínios de escritórios, shopping centers, hospitais, centros de logística, e até mesmo os edifícios residenciais de alto padrão estão adotando soluções de automação para melhorar seus serviços e elevar o seu valor de mercado. Estas soluções de automação predial incluem automação de ar condicionado, utilidades, medição e tarifação de energia, detecção e alarme de incêndio, circuito fechado de televisão, controle de acesso, sonorização ambiente e cabeamento estruturado, entre outros.

O controle de acesso, especialmente, ganhou muita importância nos últimos anos, à medida que a automação de catracas, elevadores, cancelas, fechaduras eletrônicas e outros dispositivos permitiu maior segurança, confiabilidade e menores custos operacionais.

A maioria dos edifícios corporativos possuem sistemas automatizados de controle de acesso. O mais difundido é o sistema de liberação

de catracas e portas por meio de cartão RFID, semelhantes aos cartões utilizados no transporte público. Este sistema possui vantagens como o baixo custo dos cartões, manutenção simples das leitoras de cartão e integração das informações via software, garantindo rastreabilidade e fácil gerenciamento.

Todavia, as atuais necessidades de segurança não são supridas por este método: cartões podem ser clonados ou transferidos manualmente de uma pessoa para a outra. Além disto, o cadastro de visitantes é um gargalo operacional, tendo em vista que cada novo visitante tem que ser cadastrado por um atendente em um banco de dados, incluindo muitas vezes sua foto e uma cópia digital de um documento de identificação.

Ao sair da edificação, o visitante deposita em uma urna o cartão que recebeu na entrada. Desta forma, caso o usuário retorne um outro dia, será necessário validar seu cadastro e programar um novo cartão de acesso para o mesmo.

O projeto desenvolvido permite que o próprio usuário realize seu cadastro, preenchendo seu nome, locais de acesso e tirando a sua própria

foto. Também existe a possibilidade de alterar um cadastro existente, para o caso de um operador ou gerente do sistema necessitar remover ou acrescentar permissões de acesso a um usuário cadastrado.

Além de evitar as filas e despesas operacionais de uma recepção tradicional, o sistema proporciona maior segurança, haja vista que o reconhecimento facial é verdadeiramente pessoal e intransferível. Quando o usuário precisa acessar ao sistema em uma segunda oportunidade, não é necessário refazer o cadastro. Basta se posicionar em frente a câmera e o acesso já é liberado.

O sistema implementado possui comunicação wi-fi, de modo que o reconhecimento facial pode ser feito em um local, enquanto a fechadura de acesso é acionada de forma remota.

Esta topologia torna o sistema muito mais flexível, pois uma única estação de cadastro pode controlar diversas portas à distância e de forma personalizada para cada usuário.

Responsabilidade Social do Trabalho

Um sistema de controle de acesso mais robusto permite a otimização da segurança em áreas críticas e que devem possuir acesso restrito, como datacenters, pavimentos privativos de edifícios comerciais e áreas técnicas de shopping centers, bancos e sistemas de transporte sobre trilhos, entre outros.

Agilizar o acesso de visitantes também é um ponto importante para aumentar a rentabilidade em consultórios, hospitais, casas de espetáculos e universidades.

A utilização de tecnologia sem fio permite redução de custos e mão-de-obra para a execução de infraestrutura entre a central de reconhecimento e os locais de acesso controlado. Implementar um sistema sem fio em uma edificação já existente elimina danos ao patrimônio, que seriam necessários para a instalação de um sistema cabeado.

Sustentabilidade

O projeto proporciona melhorias operacionais e econômicas para os administradores de condomínios e edifícios comerciais.

Por meio do controle adequado do acesso de pessoas não autorizadas à áreas técnicas críticas, este projeto impede que ocorram desperdícios e boicotes que podem acarretar sérias consequências financeiras e ambientais. O acesso de uma pessoa não autorizada a uma área técnica de um estação de trem, por exemplo, pode causar interrupção do serviço, depredações e utilização excessiva de transporte individual.

Outro fator positivo é a diminuição do consumo de papel para impressão de credenciais, crachás, autorizações e afins, já que o sistema é totalmente eletrônico.

Metodologia

O hardware escolhido para a estação de cadastro do sistema de controle de acesso foi o Raspberry Pi3 Model B. Trata-se de um microcomputador de baixo custo e com as dimensões de um cartão de crédito.

O projeto do Raspberry começou em 2012. A versão inicial possuía apenas um processador ARM 11 de 700MHz. Ainda assim, graças ao sistema operacional otimizado Raspbian, baseado em arquitetura Linux, o Raspberry de primeira geração já permitia um desempenho satisfatório para tarefas básicas.

Já em sua segunda versão, o Raspberry recebeu um processador de 04 núcleos e 1GB de memória RAM, além de otimizações em seu sistema operacional. Isto permitiu a aplicação do Raspberry Pi2 em diversos projetos de maior porte, inclusive na área de automação, graças as 40 conexões GPIO.

Lançado em 2016, o Raspberry Pi3 trouxe grandes melhorias, que o tornaram praticamente um computador. Em sua versão Model B, o hardware possui: Processador Quad Core com 64 bits e 1,2GHz, 1GB de memória RAM, wi-fi e bluetooth embarcados, conexão Ethernet, 40 conexões GPIO, 4 portas USB e conexão HDMI.

Para utilizar a Raspberry Pi3 como uma estação de cadastro de usuários, conectou-se à mesma um teclado USB, mouse USB e uma câmera USB. Utilizou-se também a porta HDMI para conexão com um monitor comum.

Grças à interface intuitiva do sistema operacional Raspbian, a utilização da Raspberry nesta configuração é muito semelhante a um computador tradicional.

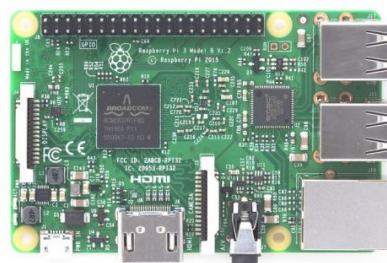


Figura 1 - Raspberry Pi3 Model B

Para a liberação dos acessos remotos, foi implementado uma conexão wi-fi entre a Raspberry e o módulo NodeMCU ESP8266.

O ESP8266 é um chip wi-fi de baixíssimo custo. Para facilitar o uso desse chip, vários fabricantes criaram módulos e placas de desenvolvimento. Estas placas variam em tamanho, número de pinos ou tipo de conexão com computador.

O NodeMCU ESP8266 é um módulo wi-fi e placa de desenvolvimento, com menos de 5cm de comprimento, que pode ser conectado diretamente a um computador por meio de uma porta micro USB e pode ser programado pela interface IDE do Arduino, que é amigável e amplamente conhecida.



Figura 2 - NodeMCU ESP8266

Estas características propiciaram grande popularidade à esta placa, principalmente entre os que buscam uma alternativa de baixo custo para aplicações wi-fi.

Esta placa possui 10 conexões GPIO, 02 push-buttons e suporte a funções como PWM, I2C e 1-wire, além de um formato especial para conexão direta ao protoboard.

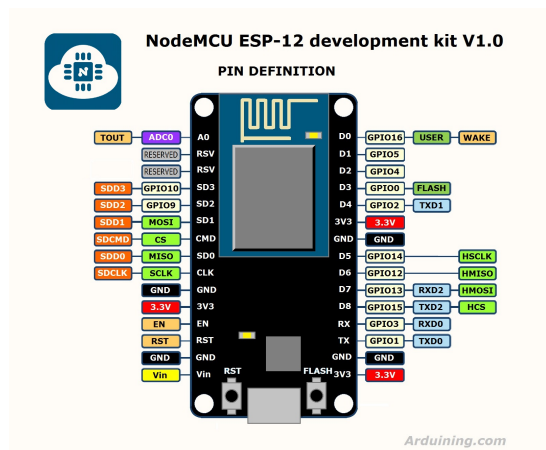


Figura 3 - Arquitetura do kit de desenvolvimento

A linguagem de programação escolhida para este projeto foi o Python. Trata-se de uma linguagem de alto nível orientada a objetos,

introduzida em 1991. Desde então, a linguagem tem sido desenvolvida de forma comunitária em código aberto, gerenciado pela organização sem fins lucrativos Python Software Foundation.

Atualmente o Python é componente de vários sistemas operacionais e aplicações, entre eles o próprio Linux.

A programação em Python é visualmente organizada, com 31 palavras reservadas e a necessidade de indentar corretamente o programa, para que o compilador possa interpretá-lo.

```
def valor1():
    while True:
        try:
            c =
            int(input('Primeiro Valor: '))
            return c
        except ValueError:
            print 'Inválido!'
```

Figura 4 - Exemplo de programação em Python

A biblioteca padrão do Python é muito extensa, incluindo aplicações para internet. Ainda assim, foi necessário acrescentar várias outras bibliotecas para permitir o reconhecimento facial, como será exposto adiante.

A programação deste projeto foi realizada no ambiente de desenvolvimento IDLE, que é distribuído e atualizado pelos próprios desenvolvedores do Python.

Uma das bibliotecas adicionais utilizadas é a OpenCV – *Open Source Computer Vision Library* (Biblioteca de Código Aberto para Visão Computacional). Esta biblioteca foi desenvolvida em 2000 pela Intel e possui código aberto e licença livre para qualquer utilização acadêmica ou comercial.

O OpenCV possui mais de 350 algoritmos de visão computacional, como filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural, entre outros. Esta biblioteca permite o processamento de imagens em tempo real, funcionalidade essencial para o reconhecimento facial instantâneo que foi proposto neste projeto.

Outra biblioteca necessária para o projeto é a *face_recognition*. Esta biblioteca foi desenvolvida a partir da biblioteca *dlib* e utiliza conceitos de deep learning (aprendizagem profunda).

Deep learning é um ramo do machine learning (aprendizado de máquina) baseado em algoritmos de modelagem. Por meio destes algoritmos, uma imagem obtida por uma câmera pode ser representada como um vetor de valores de

intensidade por pixel, ou de formas mais abstratas como um conjunto de arestas, regiões com formato particular, etc. Por meio destas representações, é possível implementar o reconhecimento facial em vídeo, como será abordado adiante.

A comunicação entre o Raspberry e o módulo wi-fi foi implementado por meio da biblioteca requests. Esta biblioteca permite o envio de requisições HTTP diretamente a partir do Python. Isto permitiu uma fácil interação entre o Raspberry e o módulo wi-fi, via endereço IP.

A interface gráfica do projeto (GUI – *Guided User Interface*) foi desenvolvida por meio da biblioteca Tkinter. Esta biblioteca permite a criação de janelas gráficas, semelhantes àsquelas encontradas em qualquer software comercial, para facilitar a interação entre o usuário e o programa. As interfaces GUI são necessárias pois a grande maioria dos usuários não possui conhecimentos para comandar o código diretamente no terminal, sendo muito mais intuitiva a utilização de “botões” que, se clicados com o mouse, levam o usuário a novos trechos do software.

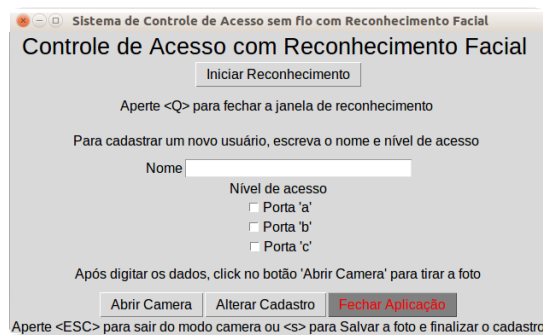


Figura 5 - Interface Gráfica do Projeto

Funcionamento do programa

Ao executar o programa, o usuário possui quatro opções, conforme demonstrado abaixo:

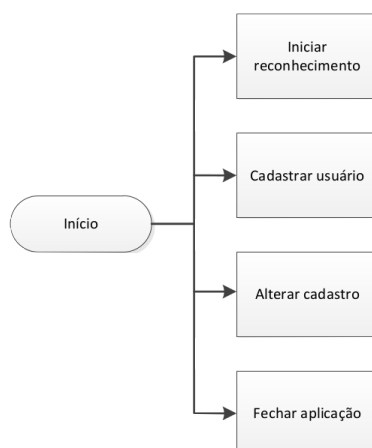


Figura 6 - Fluxograma das opções iniciais

Se o usuário já for cadastrado, ele pode optar por Iniciar o Reconhecimento diretamente, sem a necessidade de utilizar outros menus.

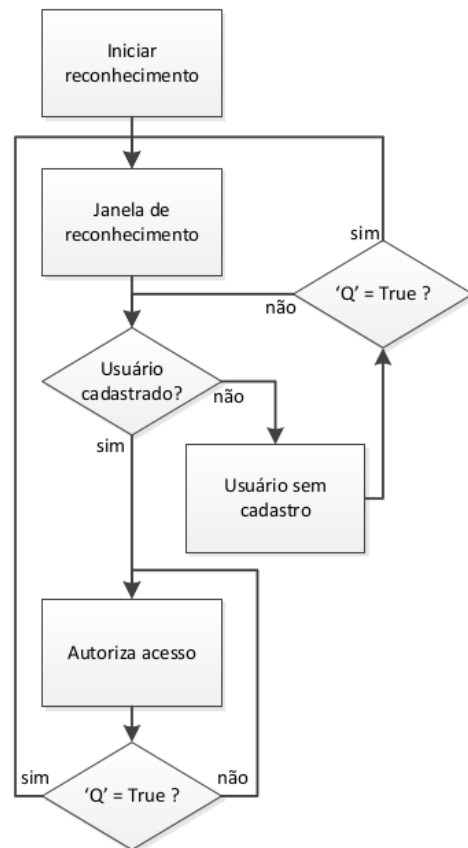


Figura 7 - Fluxograma do reconhecimento facial

Caso o usuário já possua cadastro, ou seja, já exista uma foto do mesmo no banco de dados do programa, ele terá o acesso liberado as suas respectivas portas. Caso trate-se de um desconhecido, nenhum acesso será liberado.



Figura 8 - Reconhecimento de um rosto já cadastrado

Quando um rosto já salvo no banco de dados é reconhecido, o programa gera uma saída contendo as portas que serão liberadas para este usuário.

```

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

pode abrir porta b
pode abrir porta c
Giovanni bc
pode abrir porta b
pode abrir porta c
Giovanni bc
pode abrir porta b
pode abrir porta c
Giovanni bc
pode abrir porta b
pode abrir porta c
Giovanni bc
pode abrir porta b
pode abrir porta c
Giovanni bc

```

Figura 9 - Saída do programa em Python quando um rosto é reconhecido

Independentemente do usuário ser reconhecido ou não, a janela se mantém aberta até que seja pressionada a tecla 'Q' (*quit*), que é a interrupção para esta rotina de reconhecimento.

Caso o usuário não possua cadastro no banco de dados do programa, é necessário preencher os campos destinados ao nome e as portas que poderão ser acessadas. Em seguida, ao clicar em "Abrir Câmera", uma nova janela surge. Quando o usuário pressiona a tecla "s", a foto é capturada e salva no banco de dados do programa, que localiza-se no mesmo diretório dos arquivos Python.

De agora em diante, ao acessar a interface de reconhecimento, o usuário terá, automaticamente, o acesso liberado as suas respectivas portas.

Também é possível modificar um cadastro já existente: ao clicar em "Alterar Cadastro", o usuário pode, após a confirmação do seu nome, alterar seus dados. A foto não poderá ser alterada, por motivos de segurança.

Os processos de cadastro de novo usuário e alteração de usuário existente podem ser realizados pelo próprio usuário.

Em aplicações como portarias ou recepções, é recomendável a presença de um funcionário próximo à estação de cadastro ou supervisionando em uma estação remota, para evitar que os usuários cometam irregularidades no cadastro.

Porém, em aplicações como totens para emissão de credenciais, por exemplo os totens utilizados em feiras técnicas, o usuário poderia realizar seu cadastro de forma completamente autônoma.

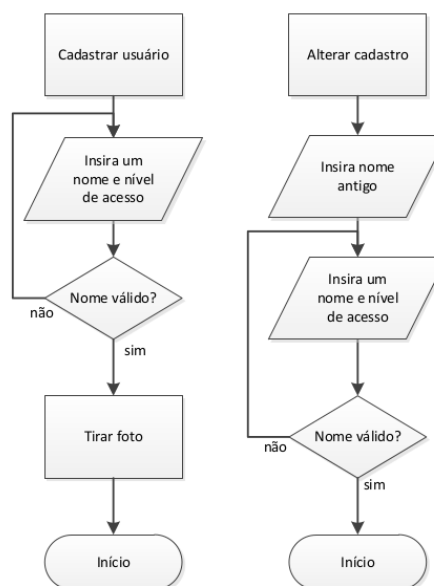


Figura 10 - Fluxograma contemplando as lógicas de Cadastro e Alteração de dados

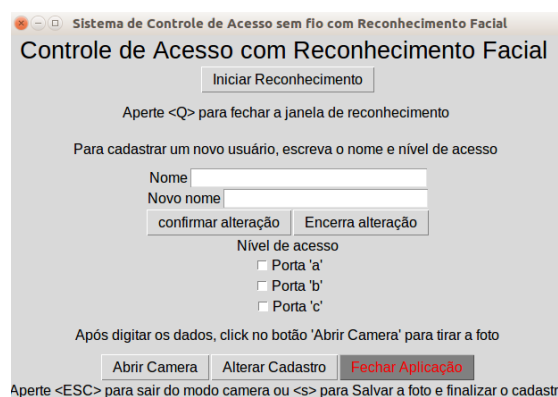


Figura 11 - Interface de alteração de dados para usuários já cadastrados

Interface com o módulo wi-fi

Quando um usuário é reconhecido e seu acesso é liberado, o programa envia dados para o módulo wi-fi remoto, usando o comando requests:

```

if (name == aba):
    hierarquia = ''
    print('Sem nível de acesso')

if ('a' in hierarquia) & (name != aba):
    print('pode abrir porta a')
    #requests.request("GET", "http://192.168.43.67:8888/gpio5on")
    #requests.request("GET", "http://192.168.43.67:8888/gpio5off")

if ('b' in hierarquia) & (name != aba):
    print('pode abrir porta b')
    #requests.request("GET", "http://192.168.43.67:8888/gpio2on")
    #requests.request("GET", "http://192.168.43.67:8888/gpio2off")

if ('c' in hierarquia) & (name != aba):
    print('pode abrir porta c')
    #requests.request("GET", "http://192.168.43.67:8888/gpio4on")
    #requests.request("GET", "http://192.168.43.67:8888/gpio4off")

```

Figura 12 - Comandos para liberação do acesso conforme as permissões do usuário reconhecido

De acordo com as portas que o usuário reconhecido está autorizado a acessar, o programa envia ao endereço IP do módulo wi-fi NodeMCU um comando para abrir a porta.

Para fins didáticos, e tendo em vista o alto custo de fechaduras eletrônicas, o módulo remoto wi-fi NodeMCU ESP8266 será montado em um painel plástico e leds serão utilizados para representar as portas.

Desta forma, haverá um led representando cada porta. Estes leds piscarão quando o acesso for liberado. Por isso, nota-se no código que para cada porta com acesso liberado há um comando “GPIO on” seguido de um comando “GPIO off”.

Caso o projeto seja implementado com fechaduras eletrônicas, basta remover o comando “GPIO off”, de modo que a fechadura permanecerá destravada até que o usuário abra a porta e acesse o local.

O módulo remoto wi-fi nodeMCU ESP8266 pode ser alimentado por qualquer conexão USB 5V. Neste projeto, a alimentação é fornecida por meio de um carregador de celular comum (fonte chaveada).

Resultados

O reconhecimento facial, isoladamente, funcionou com um grau de precisão relativamente alto desde os primeiros testes. Isto se deve ao excelente trabalho de desenvolvimento das bibliotecas `dlib` e `face_recognition`, que são públicas e recebem constantes atualizações por parte da comunidade.

Contudo, nos primeiros testes, o programa funcionava de forma muito lenta (“travando”) quando um rosto era reconhecido. Diversas otimizações foram necessárias para atingir uma fluidez adequada, especialmente na raspberry, que possui uma capacidade de processamento consideravelmente menor do que um computador tradicional.

O cadastro de novos usuários também foi uma grande dificuldade, já que as fotos de novos usuários são salvas no banco de dados e podem ser acessadas imediatamente pelo programa de reconhecimento, ou seja, a base de dados é atualizada em tempo real, tornando o programa muito mais complexo.

A interface entre o raspberry e o módulo wi-fi NodeMCU ESP8266 foi desenvolvida sem maiores problemas, graças à arquitetura extremamente amigável do módulo wi-fi, que pode ser comandado por uma única linha de código enviada ao seu IP na rede.

Por fim, a criação e customização da interface gráfica de usuário (GUI) conferiu um tom mais profissional ao projeto, tornando seu uso muito mais intuitivo mesmo para clientes sem experiência em programação.

Por meio de pesquisas e diversas tentativas, o grupo pode evoluir seus conhecimentos na linguagem Python e atingir um resultado satisfatório. O reconhecimento facial possui uma assertividade muito boa, levando em consideração que o projeto dispõe de uma webcam de baixíssimo custo. A comunicação com o módulo wi-fi é estável e possui um tempo de resposta aceitável: o intervalo de tempo entre o reconhecimento do rosto e a liberação da porta é menor que um segundo, semelhante à aplicações de grande escala.

Ocorrem alguns bugs e travamentos isolados durante a execução do programa no Raspberry. Estes não ocorrem quando o programa é executado em um computador tradicional, levando a crer que a capacidade de processamento do Raspberry Pi3 está sendo usada próximo ao limite. Isto também é evidenciado pelo próprio gráfico de “uso da CPU” do Raspberry, que constantemente atinge valores próximos de 100% durante a execução do programa.

Discussão e Conclusões

O sistema de controle de acesso sem fio com reconhecimento facial proposto se trata de uma versão acadêmica, desenvolvido com recursos limitados.

Há melhorias possíveis no sistema de segurança que são essenciais, e sem estas melhorias não seria possível aplicá-lo em escala real. Por exemplo, se um dispositivo externo obter a senha e conseguir acesso a rede wi-fi à qual estão conectados a Raspberry e o NodeMCU ESP8266, basta enviar um comando para o endereço IP do módulo wi-fi e um invasor conseguirá abrir a porta desejada. Neste caso, a vulnerabilidade do NodeMCU ESP8266 é justamente a sua extrema facilidade de operação, que permite estes comandos via web browser.

Além disto, uma foto de uma pessoa, impressa ou na tela de um celular, pode ocasionalmente ser identificada pelo programa como se fosse o próprio usuário na frente da câmera. Este problema se deve basicamente a dois fatores: a baixa qualidade da câmera utilizada no projeto e também limitações intrínsecas à biblioteca `face_recognition`, que está em desenvolvimento constante por parte da comunidade.

Contudo, este projeto prova que é possível implementar uma solução de reconhecimento facial de baixo custo, baixo tempo de resposta e alta assertividade no reconhecimento dos rostos.

Sendo assim, segue abaixo uma relação de possíveis melhorias para viabilizar o projeto em aplicações comerciais:

- Utilização de uma câmera de alta resolução, foco automático e correção de baixa

luminosidade, para dirimir os problemas de detecção de rostos em fotos ou telas de dispositivos.

- Implementação de criptografia na comunicação entre o Raspberry e o módulo wi-fi, para impedir que dispositivos de terceiros possam acessar a rede e comandar o módulo wi-fi.

- Evolução da interface do sistema operacional do Raspberry, de modo que a janela do programa de reconhecimento seja a única disponível na tela, tornando desta forma a operação do Raspberry exclusiva para o sistema de controle de acesso.

- Otimizações no programa em Python para diminuir o intervalo de tempo para abertura da câmera, bem como corrigir possíveis bugs gerais.

Após estas melhorias, o projeto estaria em condições de comercialização. Possíveis clientes seriam: recepções de edifícios comerciais, hospitais, aeroportos e universidades, entre outros.

Também é possível criar uma versão do projeto para aplicações residenciais. Neste caso, novos usuários só poderiam ser cadastrados por meio de senha. Desta forma, apenas os condôminos e pessoas de confiança teriam acesso permitido.

Agradecimentos

A Deus por nos prover saúde e força para superar as adversidades ao longo destes seis anos de graduação.

As nossas famílias, pelo amor, companheirismo, incentivo e apoio incondicional.

Aos professores Dr. Mario Cavaleiro Fernandes Garrote, Dr. Renato Camargo Giacomini e Dr. Núncio Perella, pelas colaborações diretas que tornaram possível a concepção deste projeto.

A todos os professores que compõem a FTCE, pela paciência, dedicação e ensinamentos ministrados ao longo do curso.

A Universidade São Judas Tadeu, pela excelente infraestrutura e por ceder seus laboratórios e seus materiais para a execução deste projeto.

A toda a comunidade global que trabalha, de forma anônima, para o desenvolvimento de plataformas como o Linux, o Python, e as bibliotecas utilizadas neste projeto. Estes códigos abertos proporcionam a implementação fácil e rápida de projetos inovadores e a exponencial evolução da tecnologia.

Referências Bibliográficas

1. Install Dlib on Ubuntu,
<https://www.learnopencv.com/install-dlib-on-ubuntu/>

2. Installing OpenCV in Ubuntu for Python 3,
<http://cyaninfinite.com/tutorials/installing-opencv-in-ubuntu-for-python-3/>

3. Face recognition with OpenCV, Python, and deep learning,
<https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>

4. Face Recognition,
https://github.com/ageitgey/face_recognition

5. Saiba tudo sobre o Raspberry Pi 3 e o que ele representa para o mercado,
<https://canaltech.com.br/hardware/saiba-tudo-sobre-o-raspberry-pi-3-59065/>

6. Raspberry Pi 3 Model B,
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

7. Guia do Usuário do ESP8266,
<https://www.filipeflop.com/blog/guia-do-usuario-do-esp8266/>

8. Como programar o módulo ESP8266 NodeMCU,
<https://www.filipeflop.com/blog/esp8266-nodemcu-como-programar/>

9. Python,
<https://pt.wikipedia.org/wiki/Python>

10. OpenCV,
<https://pt.wikipedia.org/wiki/OpenCV>

11. Aprendizagem profunda,
https://pt.wikipedia.org/wiki/Aprendizagem_profunda

12. Hello, Tkinter,
<http://effbot.org/tkinterbook/tkinter-hello-tkinter.htm>

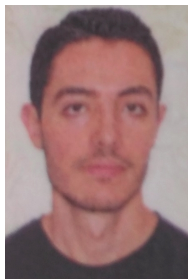
13. Using the Requests Library in Python,
<https://www.pythonforbeginners.com/requests/using-requests-in-python>

14. Interface gráfica do utilizador,
https://pt.wikipedia.org/wiki/Interface_gr%C3%A1fica_do_utilizador

15. Tkinter: Interfaces gráficas em Python,
<https://www.devmedia.com.br/tkinter-interfaces-graficas-em-python/33956>

(todos acessados pela última vez em 23/09/2018).

Dados Biográficos dos autores



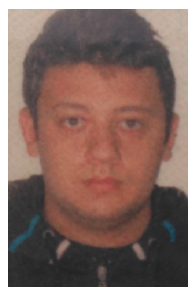
Giovanni Bruno Molitor Schiffini

10 de dezembro de 1993.
Guarulhos – SP
Colégio Torricelli
MML Sistemas de Automação
Engenharia de Aplicação e
Vendas - Técnico Orçamentista



Gustavo Mansur de Oliveira

18 de abril de 1995.
Itaquaquecetuba – SP
Sesi Jardim Monte Cristo 081
IPT – Instituto de Pesquisas
Tecnológicas
Estagiário de Engenharia



Henrique Soares Lima Costa

05 de abril de 1992.
São Paulo – SP
Colégio Augusto Maia