

# Sistema de Verificação Facial

Pedro Helias Carlos  
Universidade de Brasília - UnB  
Brasilia-DF, Brasil  
pedroheliass95@hotmail.com

Vinicius Ferreira Ribeiro  
Universidade de Brasília - UnB  
Brasilia-DF, Brasil  
ribeirovinicius08@gmail.com

**Resumo**— Sistema de verificação de faces para serviços policiais de abordagem, para identificação e consulta de ficha criminal, utilizando Raspberry Pi.

**Palavras-Chave** — *Processamento de imagem, identificação facial, Raspberry*

## I. Introdução

A tecnologia de reconhecimento facial está sendo utilizada como uma ferramenta para departamentos policiais em todo o mundo. Enquanto os softwares levantam questões sobre privacidade e uso, especialistas de segurança costumam afirmar que essa tecnologia pode ser de grande ajuda no combate ao crime. [1]

No mundo, um dos casos mais famosos sobre o uso de reconhecimento facial vem da China, onde o governo propaga uma forte vigilância: a ferramenta utilizada tem a capacidade de rastrear objetos e pessoas em tempo real, além do reconhecimento facial. [1]

Todos os dias, milhares de pessoas passam pela estação ferroviária de Zhengzhou, no leste da China. Utilizando lentes de um óculos de última geração, policiais poderão identificar, na multidão, os antecedentes criminais de quem desejarem em tempo real. Com os óculos de reconhecimento facial, cujas lentes são compostas de um vidro colorido, o policial pode tirar uma foto ou ter acesso imediato a uma base de dados sobre suspeitos. A ideia é permitir identificar mais rapidamente foragidos. [2]

Sistemas biométricos de verificação tem em sua finalidade a captação de pequenas minúcias exclusivas encontradas na formação do rosto do indivíduo, como distância entre os olhos, comprimento do nariz, entre outros. Tais informações são guardadas em um banco de dados para futuramente serem utilizadas como informação para comparação, ou em tempo real. A precisão do sistema acompanha diretamente o número de pontos específicos verificados no rosto e o equipamento utilizado. [3]

## II. Objetivos

Implementar um sistema de identificação e reconhecimento facial, que permita consultar os bancos de dados policiais, para agilizar o processo de abordagem e registro, deixando o processo

menos burocrático e mais eficiente e, por conseguinte, proporcionar atendimento mais amplo da polícia local.

## III. Justificativa

Visando implementar essa funcionalidade, foi verificada a aplicação de verificação facial para serviços policiais de abordagem. Tais serviços carecem de agilidade no processo de identificação e reconhecimento do suspeito. O projeto visa tornar esse processo mais ágil e menos custoso, além de uma ampla rede de banco de dados interligada a fim de se comparar com antigos delitos, desenhos técnicos policiais, e afins.

A utilização de uma boa câmera, para a captura da face a ser consultada, em conjunto com o bom desempenho da Raspberry Pi 3 para o processamento de imagens, deve atender a demanda do sistema proposto, que utilizará um display LCD para a exibição das informações consultadas através do banco de dados utilizado.

## IV. Requisitos

- Garantir boa nitidez das imagens;
- Verificar, com eficiência e rapidez, os traços faciais do usuário e comparar com informações previamente armazenadas;
- Construir um banco de dados inteligente que comporte ordenadamente as informações para que ocorra bons ciclos de operação do sistema.
- Necessário utilização da biblioteca OPENCV, responsável por distinguir um rosto do resto da imagem;

## V. Desenvolvimento

### A. Lista de Materiais Utilizados

Quantidade	Item	Fabricante
1	Raspberry Pi 3 - Model B	Raspberry Pi
1	Módulo Câmera V2	Raspberry Pi
1	Protoboard	Minipa
1	Display Nokia 5110	HXE

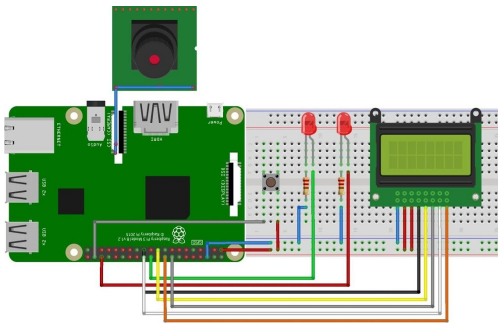
3	Leds	-
-	Jumpers	-
2	Resistores 100 k $\Omega$	-
1	PushBottom	-
1	Fonte 5 Volts	-

### B. Descrição de Hardware

A maior parte do projeto é focado em processamento de imagem, dessa forma não há uma ênfase em hardware para o projeto, a maior parte da análise é feita através do software.

O hardware é concentrado apenas para criar uma interface para o usuário, no caso o policial que está realizando o processo de identificação de suspeito, utilizando um display para exibir as informações necessárias, e a câmera utilizada para a leitura da face do suspeito.

O botão utilizado no esquemático abaixo, aciona a câmera que faz a captura da imagem para comparar com o banco de dados e identificar se o suspeito possui ficha criminal, ou não, os leds servem como recurso adicional, onde o led que é ligado pelo fio verde representa que o indivíduo não possui ficha criminal, enquanto o led ligado pelo fio vermelho, indica que o indivíduo possui ficha criminal e/ou é procurado, essas mesmas informações são mostradas no display, com o nome do indivíduo, seus documentos e sua ficha criminal.



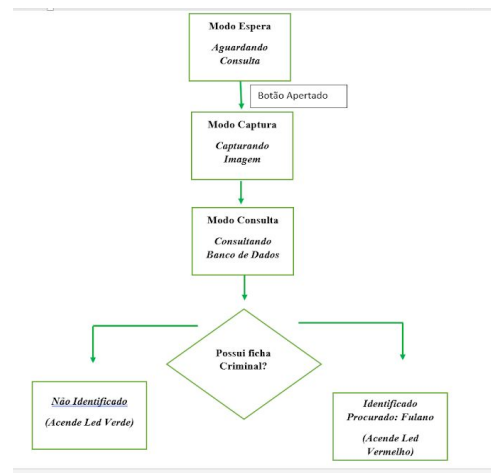
**Figura 1:** Esquemático do hardware.

### C. Descrição de Software

Para a implementação do projeto, são utilizadas as bibliotecas OpenCV[4] - Open Source Computer Vision Library, essa biblioteca tem como propósito auxiliar na construção de projetos na área de visão computacional, possuindo módulos para tratamento de imagem e vídeos, e face\_recognition [5] onde é possível fazer a comparação das faces e seu reconhecimento, além da biblioteca GPIO, para a configuração do botão, leds e display, todas utilizando a linguagem python.

Inicialmente o projeto foi implementado em linux, para verificar a funcionalidade das bibliotecas.

Com o botão sendo acionado, a imagem capturada é comparada com uma já salva anteriormente (que representa o banco de dados), carregada inicialmente junto com o código de reconhecimento. Ao iniciar o código o mesmo exibe a mensagem “*Aguardando Consulta*”, onde fica esperando o sinal enviado pelo botão, quando apertado, o programa passa a exibir “*Capturando Imagem*”, “*Consultando banco de dados*”, e caso o indivíduo não possua ficha criminal, exibe a mensagem “*Não identificado*” e acende um led verde, caso o indivíduo possua ficha criminal, exibirá a mensagem “*Identificado: “Nome do indivíduo”*”.



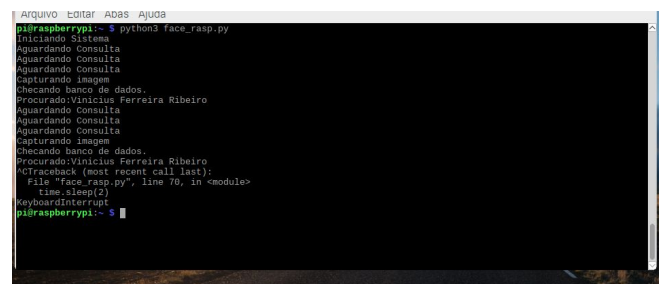
**Figura 2:** Diagrama de Blocos Do Software

## VI. Resultados

Para o ponto de controle atual, foram obtidos os seguintes resultados:

- Execução e comunicação da Raspberry e dos processos com seus periféricos (câmera e display).

Foi primeiro realizado um teste com o código de reconhecimento facial em vídeo em tempo real, para verificar se estava identificando o rosto e quem era a pessoa presente naquele vídeo, logo após, o código foi modificado para identificar através de imagem, utilizando o processo descrito na seção *Descrição de Software*, as informações com as respostas obtidas foram exibidas no prompt de comando. Foi utilizada a imagem de um dos alunos do grupo para comparação, o código utilizado e a imagem de referência, estão disponíveis no GitHub [6].



**Figura 3:** Funcionamento do projeto

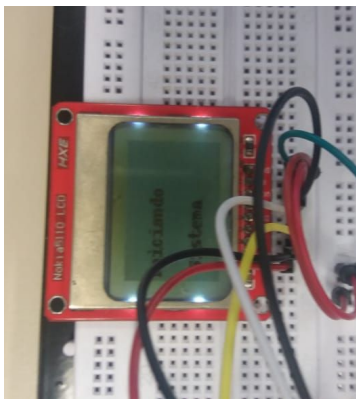
Anteriormente, encontrava-se bastante dificuldade para conduzir a comunicação entre display e Raspberry, a fim de

possibilitar a transmissão e exibição de dados via display como o planejado.

A resolução inicial do problema, a princípio, veio a partir da conversão de dados de GPIO (setmode.BCM e setmode.BOARD), possibilitando o código a se comunicar tanto com as conexões anteriores quanto com as conexões do display, originalmente BCM, assim resolvendo os conflitos das bibliotecas.

As informações exibidas no display, relacionadas às respostas obtidas através dos processos executados, foram ordenadas utilizando funções como 'clear' e setando intervalos de exibição, através de 'time.sleep()', porém, essas informações necessitam de objetos associados a imagens, assim sendo, cada imagem representa um texto.

Determinou-se os atrasos de cada função associados ao 'clear' para que a transição das imagens/textos, ocorresse da melhor forma visual possível, assim possibilitando uma boa comunicação dos processos executados, com o display [7]. Ao fim do processo, o display exibe o nome do procurado, ou, em sua ausência, exibe que o rosto analisado não representa nenhum indivíduo presente no banco de dados.



**Figura 4:** Funcionamento do display

Para o próximo ponto de controle, pretende-se ter a estrutura de adaptação necessária para melhor captura da imagem que será comparada com o banco de dados, e realizar a otimização dos processos utilizando códigos em C ou C++, conforme solicitado pelo professor.

## VII. Conclusão

Tendo em vista o conhecimento adquirido até o atual momento do curso, é possível perceber que a proposta apresentada está adequada para a disciplina, sendo sua implementação possível dentro do tempo disponível.

Chegou-se à conclusão de que para o protótipo proposto, será necessário utilizar um banco de dados reduzido, devido a complexibilidade de trabalho com grande volume de informação.

Para o próximo ponto de controle os objetivos são desenvolver um único código principal, com as subrotinas para os periféricos, utilizando linguagem C ou C++, visando sua

otimização, além de utilizar threads ou múltiplos processos na Raspberry Pi.

## Referências Bibliográficas

- [1] PAYÃO, Felipe. RECONHECIMENTO FACIAL AO VIVO É DEMONSTRADO PARA POLÍCIAS NO BRASIL. Disponível em: <<https://www.tecmundo.com.br/seguranca/130235-reconhecimento-facial-vivo-demonstrado-policias-brasil.htm>>. Acesso em: 02 Set. 2018
- [2] BBC News. OS ÓCULOS DE RECONHECIMENTO FACIAL DA POLÍCIA CHINESA QUE IDENTIFICAM SUSPEITOS EM TEMPO REAL. Disponível em: <<https://g1.globo.com/economia/tecnologia/noticia/os-oculos-de-reconhecimento-facial-da-policia-chinesa-que-identificam-suspeitos-em-tempo-real.ghtml>>. Acesso em: 02 Set. 2018
- [3] KLEINA, Nilton. COMO FUNCIONAM OS SISTEMAS DE RECONHECIMENTO FACIAL. Disponível em: <<https://www.tecmundo.com.br/camera-digital/10347-como-funcionam-os-sistemas-de-reconhecimento-facial.htm>>. Acesso em: 02 Set. 2018
- [4] GitHub OpenCV. Disponível em: <<https://github.com/opencv/opencv>>. Acesso em: 02 Set. 2018
- [5] GitHub Ageitgey. Disponível em: <[https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)>. Acesso em 10 Set. 2018
- [6] GitHub viniciusribeiro95. Disponível em: <<https://github.com/viniciusribeiro95/Sistemas-Embarcados>>
- [7] GitHub pedroheliass. Disponível em: <<https://github.com/pedroheliass/Embarcados>>.