

Ponto de Controle 2

Registrador de Presença por Detecção Facial

Leonardo Amorim de Araújo - 15/0039921 Josiane de Sousa Alves - 15/0038895

Email: leonardoaraujodf@gmail.com Email: josianealves.18@gmail.com

Universidade de Brasília

St. Leste Projeção A – Gama Leste, Brasília – DF, 72444 – 240

Resumo—Este documento apresenta uma proposta de projeto final para a disciplina de Sistemas Embarcados.

Keywords—Biometria, Detecção Facial, Raspberry Pi

I. INTRODUÇÃO

As tecnologias de reconhecimento facial não são novidade nos dias atuais. Diversos dispositivos conseguem realizar trabalhos com processamento de imagens, um exemplo são os filtros digitais para fotos, a detecção de faces em redes sociais, celulares que utilizam a face como senha pessoal e etc. Além disso, estes aparelhos que realizam processamento de imagens estão bastante acessíveis. Apesar de ser um trabalho no geral trivial para seres humanos, o reconhecimento facial é uma tarefa desafiadora para computadores. Muitas são as aplicações possíveis para a detecção facial e aproveitando-se disso será construído o dispositivo para este projeto.

II. OBJETIVOS

Construir um sistema de chamada eletrônica, com registro de presença via detecção facial utilizando a Raspberry Pi .

III. DESCRIÇÃO

É alta a demanda hoje por dispositivos que realizam detecção facial, um exemplo bem atual encontra-se na referência [1]. Uma das grandes preocupações é com a proteção de dados pessoais e com a eliminação de fraudes, que costumam acontecer ainda com o uso de senhas pessoais e cartões. Além disso, tarefas que exijam controle manual de pessoas que entram e saem em um estabelecimento, ou presença no trabalho ou em sala de aula são boas alternativas para a aplicação de projetos que envolvam biometria facial para uma possível evolução em projetos mais sofisticados, como o uso pela polícia.[3] A biometria facial é, portanto, uma alternativa para estes problemas, e com base nisto que será proposto o projeto.

IV. REQUISITOS E PROTÓTIPO

Dentre os requisitos inseridos, visa-se obter:

- Conseguir o reconhecimento de todas as faces distintas cadastradas no banco de dados;
- Ter uma câmera com capacidade de detectar a face mesmo em ambientes com luz mais baixa;
- Tempo máximo para reconhecimento da imagem e registro da presença: 5 segundos.

V. DESCRIÇÃO DE HARDWARE

Como o foco do projeto é o processamento de imagens, sua realização visa o aprendizado das funções de processamento, dessa forma, a parte de hardware é relativamente simples. Espera-se usar para a confecção do projeto a Raspberry Pi e uma câmera. Conforme a confecção do projeto for avançando, será escolhido um modelo de câmera que melhor se adapte as necessidades da proposta.

VI. DESCRIÇÃO DE SOFTWARE

Para a confecção deste projeto, está sendo utilizada a biblioteca OpenCV [4] - *Open Source Computer Vision Library*. Esta biblioteca foi desenvolvida pela Intel no início dos anos 2000 e é de uso acadêmico e comercial, para desenvolvimento de projetos na área de Visão Computacional. Possui módulos de processamento de imagens e vídeo, estrutura de dados, Álgebra Linear, GUI e muitos outros algoritmos. Inicialmente, foi desenvolvido/adaptado códigos na linguagem de programação Python para que se pudesse verificar se o projeto é factível. A figura 1 apresenta o fluxograma do projeto, onde é possível verificar a ordem de execução que os códigos que serão desenvolvidos devem ter. Abaixo será apresentado a utilidade de alguns códigos em Python que foram criados para posteriormente serem adaptados para C/C++ utilizando a Raspberry Pi.

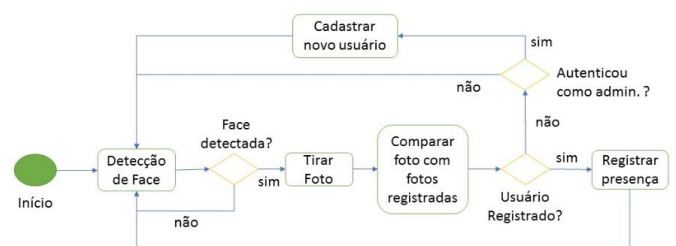


Figura 1. Fluxograma do Projeto

A. Detecção Facial

O código que realiza a detecção facial foi denominado *detect_face_from_camera.py* e utiliza um algoritmo de detecção de objetos chamados de *cascade*. Quando se consegue detectar uma face, é desenhado um retângulo no local onde está a face

da pessoa e consegue-se ainda verificar onde estão os olhos, com outro *cascade* de reconhecimento. A detecção facial será interessante para o projeto pois após a verificação de uma face com a câmera, será tirada uma foto, o que será explicado no tópico a seguir.

B. Tirar e Salvar Fotos

Foi criado também um código que consegue tirar fotos com a câmera principal do computador e salvá-las. Este código foi denominado *take_picture.py*. No código, ao pressionar a tecla *SPACE*, uma foto é tirada e ao pressionar a tecla *ESC*, a execução é interrompida, terminando o programa. Apesar de ter utilizado o mecanismo de pressionar teclas para tirar fotos, a ideia é que este programa consiga se comunicar com o programa anterior de detecção facial, de modo que ao se detectar uma face, uma foto seja tirada. Uma vez que um rosto for detectado, uma foto será tirada automaticamente após um período (2 segundos em média). A foto capturada será salva (também automaticamente) no diretório padrão onde o programa estiver instalado. Com a imagem salva, será feita a verificação conforme explicado abaixo.

C. Verificação da Foto Tirada

Dois outros códigos que podem ser utilizados como base foram chamados de *read_image_1.py* e *read_image_2.py*. O primeiro, utiliza as funções da biblioteca OpenCV para abrir a imagem e o segundo abre a imagem com o módulo matplotlib do python. Nesses códigos não foi utilizado nenhum algoritmo para comparação de imagens, pois isto será desenvolvido no decorrer dos próximos pontos de controle. A imagem salva será comparada a uma série de fotos armazenadas em um banco de dados. Se com essa comparação for detectada uma similaridade grande (maior ou igual a 70%, por exemplo), será registrada presença para o aluno cuja foto armazenada no banco de dados apresentou similaridade com a imagem capturada. Caso a comparação não detecte similaridade com nenhuma imagem armazenada no banco de dados, será mostrada uma mensagem de erro, permitindo ao usuário a opção de capturar uma nova foto ou cadastrar a foto tirada no banco de dados.

D. Registro da Presença

Nesta etapa, já que foi verificado que uma pessoa quer registrar sua presença e está cadastrada no banco de dados, um programa irá criar ou abrir um novo arquivo para que seja computado o nome, matrícula e a hora da presença da pessoa, assim como a data. Se possível. Esses dados podem ser disponibilizados em uma planilha no *Google Drive*, através do método *sockets*, aprendido em sala de aula.

E. Cadastro de Novo Usuário

Caso na etapa da verificação a foto não corresponda a nenhum dado no sistema, o usuário terá a opção de tirar uma nova foto ou cadastrar a foto capturada, como já citado acima. Caso a segunda opção seja escolhida, dados do novo usuário deverão ser fornecidos, como nome, matrícula e cadastro da nova foto. O cadastro deverá ser autorizado pelo administrador do sistema (root).

VII. RESULTADOS

Para este ponto de controle, os resultados obtidos foram:

- Um código para abertura de imagens;
- Um código para reconhecimento facial;
- Um código para tirar uma foto quando uma tecla do teclado é pressionada.

Um dos objetivos para o próximo ponto de controle, é conseguir colocar todas essas funções em um único código, além de transferir esse novo código para a linguagem C ou C++. Ainda para o próximo ponto de controle, espera-se conseguir utilizar as ferramentas aprendidas em sala de aula para manipulação de processos, sinais, threads, etc.

VIII. CONCLUSÃO

Tomando como base os conhecimentos adquiridos até o presente momento na disciplina de Sistemas Embarcados, bem como o plano de ensino para o decorrer do semestre, é possível concluir que a proposta de projeto apresentada está dentro do esperado para a disciplina, sendo possível a sua conclusão no tempo disponível para a sua execução.

IX. ANEXOS

A. Códigos

Todos os códigos podem ser visualizados no link para o github a seguir: https://github.com/leonardoaraujodf/Sistemas_Embarcados/tree/master/2_PC/codigos_pc2

REFERÊNCIAS

- [1] Biometria facial começa a ser testada no transporte público do DF, acesso em 04/10/2017. http://www.correiobraziliense.com.br/app/noticia/cidades/2017/05/16/interna_cidadesdf,595226/biometria-comeca-a-ser-testada-no-distrito-federal.shtml
- [2] Aprenda a desbloquear seu notebook por reconhecimento facial, acesso em 04/10/2017. <http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2015/06/aprenda-desbloquear-seu-notebook-por-reconhecimento-facial.html>
- [3] 100+ Projects in Image Processing and Fingerprint Recognition, acesso em 04/10/2017. <http://projectabstracts.com/list-of-projects-on-image-processing>
- [4] OpenCV library, acesso em 04/10/2017. <http://projectabstracts.com/list-of-projects-on-image-processing>