Reconhecimento de Faces e Identificação de Regiões de Interesse em Cenas para o Acompanhamento Baseado na Movimentação de Servomotores e Inteligência Artificial

Erick O. Santos[1], **Luiz C. C. Moitinho**[1], **Wedson T. de Almeida**[1], **Alcides X. Benicasa**[1]

[1]Departamento de Sistemas de Informação – Universidade Federal de Sergipe (UFS)
Caixa Postal – 49.506-036 – Itabaiana – SE – Brasil

{elderick,luizmoitinho,wedson}@academico.ufs.br,alcides@ufs.br

Abstract. *With the evolution of the speed of computers and the cheapening of eletronic components, through eletronic prototype devices, arives the oportunity of creating systems that aim to cheapen and increase the availability of facial recognition solutions, added to the use of microcontrollers and servo motors, appropriating of smartphones as the bridge to provide the service to the end user in a real time performative application.*

Resumo. *Com a evolução da velocidade dos computadores e o barateamento de componentes eletrônicos, mediante aos dispositivos de prototipagem eletrônica, surge a oportunidade de criar sistemas que visem baratear e aumentar a disponibilidade de soluções de reconhecimento facial, aliado a utilização de microcontroladores e servomotores, apropriando-se de smartphones como ponte para disponibilizar o serviço para o usuário final em uma aplicação performática em tempo real.*

1. Introdução

Ao passo em que as tecnologias e a automação evoluem, os problemas que decorrem delas se tornam mais complexas, dessa forma, se têm necessária a criação de soluções que possam ser autônomas e/ou controladas à distância para a realização de atividades e processos. Diante disto, surgiram situações recorrentes que levaram à adoção crescente de módulos inteligentes capazes de realizar comunicações entre diversas aplicações, bem como dispositivos de hardware, a fim de propiciar soluções que sejam cibernéticas.

No trabalho de [Braga and Costa 2015], por exemplo, foi proposto um modelo para a movimentação autônoma de um robô a partir da detecção facial e consequente movimentação em direção ao alvo. Em [Oliveira and Moreira 2021] foi proposto um sistema capaz de detectar a sonolência do motorista por meio da captura e processamento digital das imagens em tempo real, gerando como saída a emissão de alertas sonoros para que o motorista possa despertar.

Buscando reproduzir a natureza humana, estudos e técnicas vêm sendo empregadas na computação, dentro das área de Inteligência Artificial (IA) e Internet das Coisas (IoT - *Intelligence of Things*), capazes de fornecer aos dispositivos programados comportamentos inspirados nos sistemas motores e neurológicos como, por exemplo, o controle e movimentação do corpo humano, orientação do campo de visão, detecção e reconhecimento de regiões de interesse, consolidados por meio da junção de componentes de hardware e software.

Tendo em vista a visão computacional, o desenvolvimento de equipamentos computacionais cada vez mais rápidos, permitiram muitos avanços perceptíveis nesta área nos últimos anos. Atualmente, sistemas de reconhecimento de faces a partir de imagem de vídeos ou mesmo imagens estáticas tem se tornado uma realidade. As aplicações dessas pesquisas vão desde o controle de acesso a prédios e bancos até sistemas de reconhecimento de faces acoplados em robôs que, inclusive possuem a capacidade de perceber o estado emocional do seus operadores [Lopes 2005].

2. Revisão de Literatura

Na presente seção serão apresentados conceitos e métodos para embasamento e desenvolvimento do trabalho, englobando uma visão sobre detecção de movimento e face, centralização de face em cena e reconhecimento facial.

2.1. Detecção de Movimento

A implementação da detecção de movimento pode ser feita através de duas técnicas principais, a subtração de fundo e a diferença temporal entre imagens [Hu et al. 2004]. A subtração do fundo consiste na diferença espacial entre o modelo do ambiente e a imagem atual captada. A diferença temporal entre imagens consiste na subtração dos *pixels* de quadros sucessivos, o qual foi utilizado no projeto.

De acordo com [Ferreira 2012], o algoritmo de subtração dos pixels de quadros consecutivos pode ser dado pela Equação 1, descrita como segue:

$$d(x,y,y) = \begin{cases} 1 \text{ se } |f(x,y,t) - f(x,y,t-1)| > T_d \\ 0 \text{ os pixels restantes} \end{cases} \quad (1)$$

onde T_d é um limiar predeterminado, $f(x,y,t)$ é uma imagem tomada no tempo t e $f(x,y,t-1)$ é uma outra imagem da sequência em um tempo anterior, geralmente de um a três quadros anteriores. O $d(x,y,t)$ é a imagem de movimento, onde todo pixel com valor "1" é considerado como parte de um objeto em movimento.

2.2. Detecção e Reconhecimento Facial

Considerada como uma importante etapa para o desenvolvimento deste trabalho, a detecção facial, de acordo com [Schmidt and Nogueira 2016], consiste em identificar e isolar a área referente à face em uma imagem digital, e pode ser realizada com base em vários atributos: formato do rosto ou cabeça, aparência da face, ou a combinação destes. A maior parte dos algoritmos de detecção facial são os baseados em detectar o formato do rosto e também de extrair as informações do mesmo, como olhos, nariz, boca, entre outras.

Após ser aplicado o algoritmo de detecção facial, tendo utilizado como entrada imagens processadas e convertidas em escala de cinza, e uma determinada face ser detectada, deverá ocorrer o processo de segmentação da imagem, ou seja, a separação da área de interesse, que nesse caso é a face, descartando o restante da imagem. O algoritmo utilizado para detecção facial foi o Viola-Jones, algoritmo proposto por [Viola and Jones 2001] que é um classificador do tipo *Haar*. Esta técnica de detecção de padrões consiste em encontrar partículas de imagem que apresentem diferença de contraste. Esses quadros (contrastes) são denominados características de *Haar*.

A aplicação do detector é realizada de forma integral, deslizando as características sob a imagem, computando as médias de valores dos *pixels*. Se a diferença entre as áreas estiver abaixo de um limiar pré-determinado, a característica coincide [Paixao 2018]. Então, esse resultado irá representar o valor encontrado pela característica para determinada região.

De acordo com [Viola and Jones 2004], o princípio do funcionamento deste detector é chamado de classificador em cascata, que consiste na técnica de agrupar, por estágio, outros classificadores para conseguirem altas taxas de detecção e, então, determina que a avaliação de um segundo classificador só será invocada caso a avaliação do primeiro seja positivo. Caso contrário, o procedimento é interrompido e a sub-janela é rejeitada.

Outra etapa importante considerada aqui é reconhecimento facial, tratando-se de uma categoria de segurança biométrica baseada na utilização de medidas biológicas ou físicas para identificar um indivíduo. O reconhecimento facial pode ser utilizado para diversas tarefas como, por exemplo, desbloquear telefones, fazer cumprir a lei, reduzir o crime no varejo, marketing, reconhecimento de motoristas, entre outras aplicações. O funcionamento do reconhecimento facial depende de um conjunto de áreas do ramo da Inteligência Artificial, como o *Machine Learning* e o *Deep Learning* [Karspersky 2021].

Existem várias formas de realizar o processo de reconhecimento facial. De acordo com o trabalho de [Geitgey 2016] e descritos nos parágrafos a seguir, um *pipeline* básico de reconhecimento facial geralmente consiste de 4 passos: encontrar todas as faces em uma imagem, posicionar e centralizar as faces, gerar as medidas da face, comparar as medidas da face com todas as outras pessoas já reconhecidas e determinar quem é o indivíduo.

Após a detecção da face é possível realizar reconhecimento facial, o qual será necessário encontrar uma segmento da imagem que se assemelhe ao padrão HOG (do inglês *Histogram of Oriented Gradients*). Um padrão HOG é um modelo extraído de outros grupos de faces de treinamento. Caso um padrão similar seja encontrado na imagem, consideramos que ali existe uma face, então, obtemos as coordenadas da face na imagem.

A segunda etapa do *pipeline* de reconhecimento facial é realizar a centralização dos olhos e boca. Esse processo é realizado para simplificar o processo de reconhecimento para o caso de faces de uma mesma pessoa que estejam em ângulos diferentes. Nesta etapa, é utilizado um algoritmo denominado *face landmark estimation*, ou, estimativa de marcas do rosto. A ideia básica do algoritmo é a de que existem 68 pontos específicos que existem em todas as faces, então, um algoritmo de machine learning é treinado para conseguir encontrar todos esses pontos em uma face.

A terceira etapa do processo é codificar as faces encontradas. Para isso, é necessário extrair uma série de medidas de cada face. Porém, as medidas que podem fazer sentido para um humano, como cor dos olhos ou tamanho do nariz, podem não fazer sentido para um computador, então, pesquisadores descobriram que uma abordagem muito precisa é utilizar *deep learning* e deixar que o computador decida quais medidas são mais importantes e às colete. Uma rede neural é então treinada para gerar 128 medidas para cada face.

Quanto à última etapa, segundo [Navlani 2018], para realizar a busca de reconhecimento da face podem ser utilizadas diversas técnicas. Uma técnica de busca famosa

e que foi utilizada durante o desenvolvimento deste projeto através da extensão cube do PostgreSQL é a K-Nearest Neighbors (KNN). No KNN a estrutura do modelo é definida pelo conjunto de dados, no caso, a estrutura do modelo é representada pelas 128 medidas geradas da face. Consideramos K como a quantidade de vizinhos próximos, quando K é igual a 1, o algoritmo só levará em consideração o vizinho mais próximo, obtendo então a face com maior similaridade dentre as existentes na base de dados.

2.3. Centralização de Face

Considerado como um dos principais componentes eletrônicos presentes no uso de prototipagem eletrônica e também de acordo com [Junior 2011], os servomotores são componentes em que o controle de velocidade e rotação são controlados através de realimentação de um sensor acoplado ao eixo e controlador específico, podendo ser classificado como motores síncronos.

De acordo com [Tannus 2018], o servomotor é formado por uma combinação de quatro componentes: um motor DC, engrenagens, circuito de controle e um potenciômetro. O circuito de controle (juntamente com o potenciômetro) forma um sistema de retroalimentação, o qual é capaz de controlar de forma mais precisa possível a posição do eixo do motor. Os servomotores tradicionais possuem sua rotação limitada entre 0 e 180 graus, sendo que esse valor angular é definido pela entrada de controle do dispositivo, uma vez que o valor de entrada é alterado, o posicionamento do potenciômetro é alterado para refletir esta alteração. Destacamos aqui seus principais componentes:

- **Circuito de Controle**: responsável por monitorar a atual posição do potenciômetro de acordo com o sinal do receptor e a posição do mesmo. Também é responsável por receber os sinais e energia do receptor;
- **Motor**: movimenta a engrenagem e o eixo principal do servomotor;
- **Engrenagens**: contribui para a diminuição de rotação do motor, direcionam mais força ao eixo principal de saída e movimentam o potenciômetro e eixo; e
- **Potenciômetro**: integra o eixo de saída e fiscaliza a posição do servomotor.

Conforme apresentado e também de acordo com [Tannus 2018], o controle de posicionamento do servomotor é realizado mediante a um sinal modulado por largura de pulso PWM (do inglês *Pulse Width Modulation*), tendo um circuito de controle que é responsável por monitorar o sinal de entrada a cada 20 milissegundos (ms), verificando alterações de 0V e 5V em um intervalo de 1 ms e 2 ms. Estas alterações designam uma variação na posição do eixo e/ou movimentação do servo. Sendo que o valor de 1 ms indica o início da faixa de valores, ou seja, 0 graus. Por outro lado, um período de 2ms indica que o motor está em 180 graus.

3. Desenvolvimento

Nesta seção será apresentada a arquitetura proposta e desenvolvida neste projeto.

Podemos observar na Figura 1 o modelo geral deste projeto, dividido em três componentes principais: uma aplicação cliente na plataforma de prototipação ESP32-CAM, um servidor (API) *Python* [Grinberg 2018] e uma aplicação móvel. O ESP32-CAM é responsável pela captura e envio de imagens, realizar cálculos e pelo controle e reposicionamento dos servomotores que, para este fim, utiliza-se de coordenadas enviadas pelo servidor Python. No servidor *Python* serão realizados os seguintes processos:

- Detecção de movimento;
- Identificação da localização das faces detectadas;
- Envio das coordenadas para reposicionamento dos servomotores;
- Armazenamento e reconhecimento das faces detectadas (conhecidas e desconhecidas);
- Envio de notificação de reconhecimento em tempo real e;
- Disponibilização de serviços de listagem de imagens para a aplicação *mobile*.

Já o ESP32CAM é responsável por:

- Captura e transferência da imagem para a API *Python*;
- Realizar cálculos de variação de ângulo dos servomotores e;
- Centralização do alvo em cena, reposicionando os servomotores.

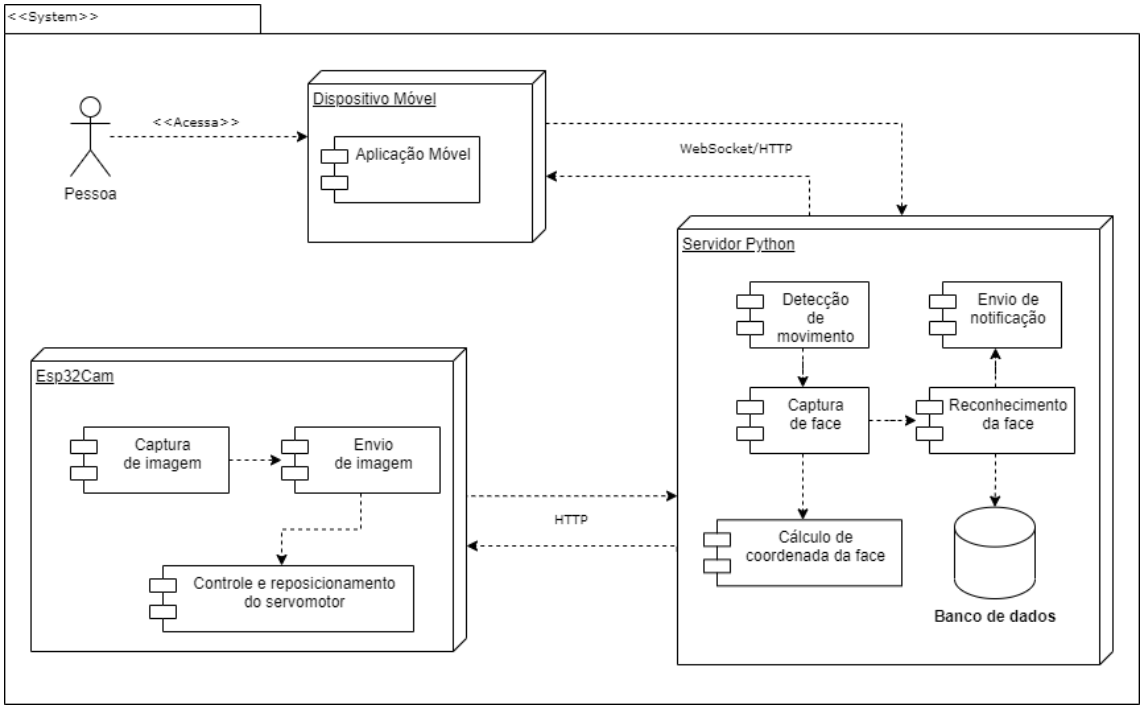

Figura 1. Arquitetura do sistema.

3.1. Captura e Transferência da Imagem

Com a câmera devidamente configurada, é iniciado um laço de repetição (*loop*), responsável por enviar uma foto para o servidor a cada intervalo de tempo. O fluxo principal funciona da seguinte forma, é feito a captura da câmera no dado momento, a imagem é transformada em $base64$, o cabeçalho de requisição HTTP é montado e em seguida a foto é enviada para o servidor.

3.2. Detecção de Movimento e Face

Para a detecção de movimento foi aplicado o algoritmo de análise temporal de quadros consecutivos, este processo é baseado na comparação das três últimas imagens. As comparações são realizadas mediante a função *absdiff* da biblioteca *OpenCV*, que recebe dois vetores (uma imagem é um vetor bidimensional) como parâmetro e retorna a

diferença absoluta que existe entre os pares de imagem. O método de comparação recebe dois vetores (uma imagem é um vetor bidimensional) como parâmetros e retorna a diferença absoluta que existe entre os dois. Após concluída a comparação, é realizada a operação and, bit a bit entre os pixels das imagens. Essa operação tem como objetivo retornar a intersecção entre os pares de imagens.

Após o retorno da operação de intersecção, é aplicada uma máscara sobre a imagem, a qual tem como objetivo obter uma imagem binária, onde é atribuído valores aos *pixels* da imagem em relação ao valor de limite fornecido, ou seja, cada valor de *pixel* é comparado com o valor do limiar, caso o valor do *pixel* seja menor que o limite, ele será definido como 0, caso contrário, será definido como um valor máximo que foi definido. Ao final, a imagem ficará padronizada com pixels com valor 0 ou com o valor limite definido, auxiliando assim no cálculo da detecção de movimento.

Após a detecção de movimento, caso positivo, é carregado o classificador em cascata e aplicado sobre a imagem. De um modo geral, o processo consiste em retornar uma estrutura de dados com quatro valores para cada face, necessários para formar os pontos de extremidades do recorte da face. Com isto, é possível descobrir a localização de cada face, e assim ser capaz de enviar esses dados para o módulo de reconhecimento, responsável por realizar reconhecimento facial dos segmentos de faces encontrados.

Levando em consideração que em uma imagem possa existir mais de uma face, e ainda considerando que um servomotor (hardware responsável pela movimentação) se limita a centralização apenas de um ponto, foi necessário identificar a maior face, tomada aqui como a mais importante. A localização é baseada pelo tamanho, ou seja, a face que conter uma maior quantidade de pixels é eleita como a face evidente.

Após encontrada a face em evidência, se faz necessário converter seus pontos em apenas um ponto, que representará o ponto central da face. Posteriormente ao cálculo das coordenadas da posição central da face em evidência, as coordenadas são enviadas para o módulo de centralização da face, que, mediante a resposta da requisição, reposicionará os servomotores.

3.3. Variação Angular e Reposicionamento dos Servomotores

De modo elucidativo, o início dos procedimentos para a movimentação dos servomotores se dá no recebimento dos valores finais que os servos devem assumir ao final do processo. Baseando-se em um plano cartesiano, foi possível mapear as movimentações dos servos nos sentidos horizontal e vertical, sendo equivalente às coordenadas dispostas em termos de x e y. Por fim, para a centralização do alvo, primeiramente faz-se necessário realizar um cálculo para encontrar o ângulo do local de atenção atual até a coordenada recebida, que é realizada mediante o cálculo da distância euclidiana entre dois pontos. Adotamos neste trabalho a distância bidimensional, dada por $P = (px, py)$ e $Q = (qx, qy)$, calculada como segue:

$$D = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \tag{2}$$

Este cálculo será aplicado entre três pontos, A (local de atenção atual), B (posicionamento da câmera) e C (coordenada recebida referente ao alvo), sendo representandos por a, b e c, expresso pelas seguintes equações:

$$a = \sqrt{(B_x - C_x)^2 + (B_y - C_y)^2} \tag{3}$$

$$b = \sqrt{(A_x - C_x)^2 + (A_y - C_y)^2} \tag{4}$$

$$c = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2} \tag{5}$$

Com as distâncias calculadas é possível obter o ângulo entre o local atual e o alvo, expressado por:

$$\alpha = \sqrt{1 - (\frac{b^2 + c^2 - a^2}{2 * b * c})^2} \tag{6}$$

$$\theta = \frac{\alpha * 180}{\pi} \tag{7}$$

$$f(\theta) = \begin{cases} |\theta| * -1 & \text{se} \quad Cx > Bx \\ |\theta| & \text{se} \quad Cx < Bx \end{cases} \tag{8}$$

Após encontrar o ângulo entre o local de atenção atual até o alvo, tido em $f(\theta)$, tal valor é passado como parâmetro para a função responsável por realizar seu mapeamento para os limites de movimentos definidos para os servos. De maneira geral, mapeando um número de um intervalo para outro, ou seja, equipara valores dentro de uma faixa para valores dentro da outra faixa, retornando o valor real a ser aplicado para realizar a movimentação do servomotor.

3.4. Reconhecimento Facial e busca

O processo de reconhecimento facial se utiliza de um banco de dados para armazenar e buscar características das faces capturadas, para então determinar se o indivíduo é ou não conhecido. De qualquer forma, o resultado da classificação é enviado, via notificação em tempo real, para a aplicação mobile.

Primeiramente, dado o recorte de uma face, é utilizado um método que confirma a existência de uma face na foto. Caso não seja detectada uma face na imagem, o processo de reconhecimento é interrompido. Com a confirmação da existência da face, é realizado o processo de reconhecimento facial, que consiste na geração das medidas faciais utilizando uma rede neural pré-treinada e a busca por faces similares em banco de dados de medidas faciais. Caso a busca retorne um identificador de uma pessoa, significa que o processo considerou que esta possui uma face similar o suficiente para que sejam consideradas a mesma pessoa, então se insere a face recém chegada relacionando-a com a pessoa obtida. Caso a busca não retorne nenhuma informação, significa que nenhuma pessoa armazenada foi considerada similar o suficiente, então é criada uma nova pessoa e a foto recém chegada é relacionada com a pessoa criada. Ambos os fluxos realizam o envio da notificação em tempo real para a aplicação mobile, que informa o nome do indivíduo, caso o mesmo seja considerado conhecido para o usuário da aplicação, ou uma notificação genérica de reconhecimento, caso seja considerado desconhecido.

4. Resultados

Em um primeiro momento, foi realizado um teste no período noturno, contando apenas com as luzes internas do ambiente. Como é visto na Figura 2, os servos se encontram na posição de 90º quando o ESP32CAM capturou uma imagem e por seguinte a enviou para o serviço *Python*, que fez o processamento de detecção de face, demarcou a imagem em termos dos eixos x e y e capturou as coordenadas da mesma, tendo como resultado a Figura 3.

Figura 2. Captura da imagem inicial.

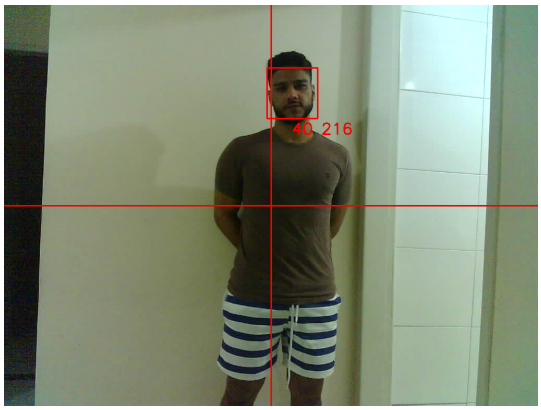

Figura 3. Detecção da coordenada da face.

Após a obtenção da coordenada em que se encontra a face detectada, o serviço *Python* retorna este valor para o ESP32CAM que, por sua vez, realiza o processo de leitura e validação da mesma para que seja realizada a variação das posições dos servomotores. Em seguida, é realizado o cálculo para encontrar o ângulo do local de atenção atual até a coordenada recebida e o mapeamento dos valores para os valores limítrofes dos servomotores, baseando-se em seus coeficientes, para que então seja realizado o reposicionamento dos servomotores para a centralização do alvo na cena. Após isto, o ESP32CAM captura novamente uma imagem com o alvo centralizado (Figura 4) que, ao ser recebida no serviço *Python*, tem a imagem marcada em termos dos eixos x, y e a coordenada da face detectada, confirmando a centralização do alvo em cena (Figura 5).

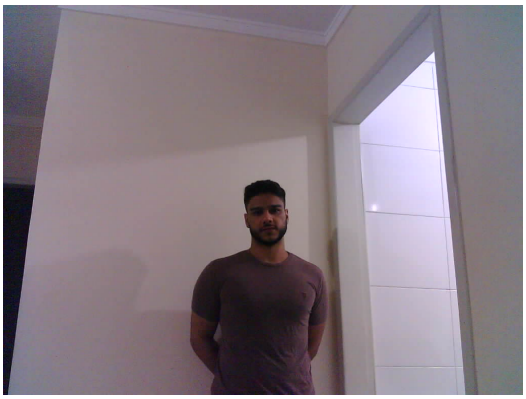

Figura 4. Captura após a centralização.

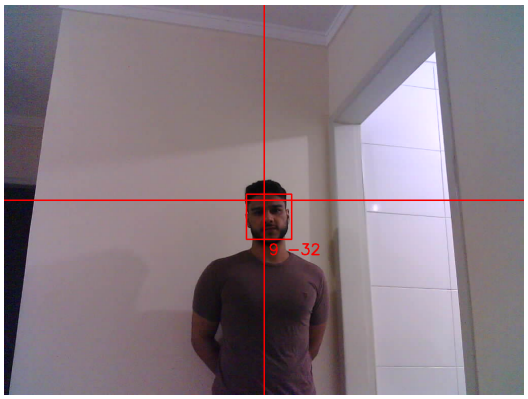

Figura 5. Face detectada e centralizada.

A partir disto, o processo de reconhecimento facial é realizado. Considerando que o banco de dados esteja vazio, o processo de relacionamento de entidades tomado após a busca é o de criação de uma nova pessoa, e sua referenciação na tabela do banco de dados com nome de ”Foto”correspondente a inserção da face. A notificação de reconhecimento é então enviada para a aplicação *mobile*, que exibirá a notificação e a face recém reconhecida (Figura 6).

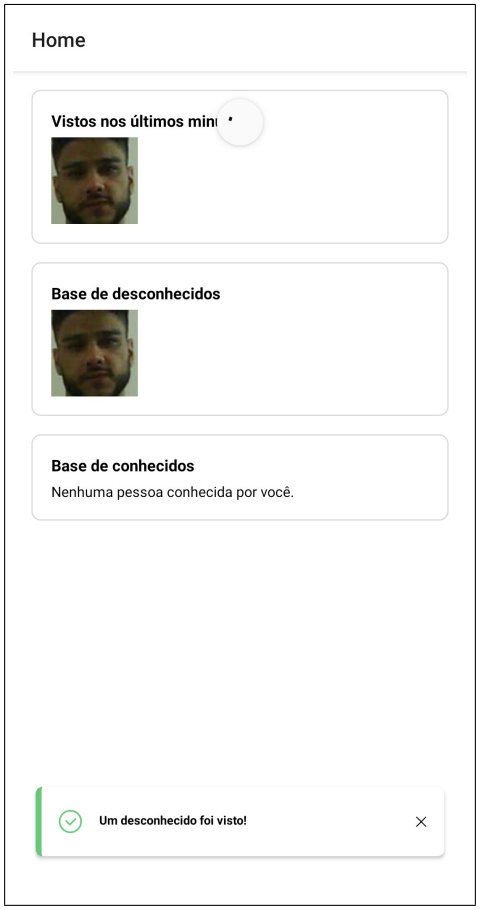

Figura 6. Notificação de reconhecimento recebida.

5. Considerações finais

Como forma de sofisticar o trabalho, é viável explorar outros métodos de aperfeiçoamentos relacionados à captura da cena em tempo real, bem como o estudo sobre a utilização de outras técnicas para detecção de movimento e face, comparado aos resultado já obtidos.

Há ainda, outros fatores a serem refinados no âmbito dos dispositivos de prototipagem, em específico a movimentação do servomotores, bem como o aprimoramento do controle do passo dos mesmos e do algoritmo de centralização do alvo. Outro fator é, adicionar o recebimento de um terceiro argumento (além das coordenadas), capaz de informar uma taxa de distância a qual a face foi detectada em uma imagem, para que assim o servomotor possa ajustar o quão deve variar o seu passo ao longo da mesma.

Por fim, podem ser exploradas novas formas para o aperfeiçoamento entre as en-

tidades no processo de reconhecimento facial, bem como a definição de um limite de armazenamento de fotos utilizadas no reconhecimento para cada pessoa no banco de dados, buscando salvar apenas as melhores amostras disponíveis para um indivíduo.

6. Referencias

Referências

Braga, E. S. and Costa, R. M. R. (2015). Imagem e movimento: implementando o movimento autônomo através do arduino e opencv. *Centro de Ensino Superior de Juiz de Fora (CESJF)*.

Ferreira, C. S. (2012). Implemantação do algoritmo de subtração de fundo para detecção de objetos em movimento, usando sistemas reconfiguráveis. *Universidade de Brasília*.

Geitgey, A. (2016). Machine learning is fun! part 4: modern face recognition with deep learning. *Medium. Medium Corporation*, 24.

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). A survey on visual surveillance of object motion and behaviorsh. *IEEE Trans. on Systems, Man and Cybernetics*, 3:334–352.

Junior, G. C. d. N. (2011). *Máquinas elétricas: teoria e ensaios*. Saraiva Educação SA.

Karspersky (2021). What is facial recognition – definition and explanation. Disponível em: `https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition`. Acessado em: 16 de julho de 2021.

Lopes, E. C. (2005). Detecção de faces e características faciais. *Pontifícia Universidade Católica do Rio Grande do Sul*.

Navlani, A. (2018). Knn classification using scikit-learn. Disponível em: `https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn`. Acessado em: 16 de julho de 2021.

Oliveira, K. R. and Moreira, W. D. P. (2021). Sistema de detecção de sonolência em motoristas. *Pontifícia Universidade Católica de Goiás*.

Paixao, E. P. (2018). Aplicação do algotitmo viola-jones na detecção de objeto. *Universidade Regional do Noroeste do Estado do Rio Grande do Sul – NIJUÍ*.

Schmidt, A. E. and Nogueira, E. C. (2016). Estudo sobre métodos de reconhecimento facial em fotografias digitais. *A Mostra Nacional de Iniciação Científica e Tecnológica Interdisciplinar (MICTI)*.

Tannus, A. M. (2018). Arduino: Servomotores. *Centro Universitário de Anápolis Unievangélica*.

Viola, P. and Jones, M. (2001). rapid object detection using a boosted cascade of simple features. *IEEE Computer Society Conference*.

Viola, P. and Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*.