Sistema embarcado para o controle de acesso em áreas restritas de prédios inteligentes

Hyuri S. Maciel, David F. Silva, Clóvis G. M. do Nascimento e Andre L.L Aquino

[1] Instituto de Computação – Universidade Federal de Alagoas – Maceió – AL – Brasil

{smhyuri, firminosilva.david, clovisgabrielm, alla.lins}@gmail.com

Abstract. *This paper presents an embedded system for access control in restricted area through the integration of two authentication strategies: facial recognition and RFID TAGs. Once that face recognition, in an embedded system, is susceptible to failures and RFIDs TAGs can be used by other users, the integration of these two solutions aims to increase the robustness of access control system. This system can be easily deployed into enterprise environments, meetings and offices rooms. Additionally the system proposed does not require the need to have the system connected to the Internet. Through the evaluations we observed an acceptable execution time and better robustness in access control system, even using simple techniques of face recognition.*

Resumo. *Este trabalho apresenta um sistema embarcado para o controle de acesso em áreas restritas por intermédio da integração de reconhecimento facial e autenticação com RFID. A escolha da integração das duas soluções tem como objetivo aumentar a robustez do sistema para o controle de acesso, uma vez que o reconhecimento facial, quando concebido num sistema embarcado, é suscetível a falhas no reconhecimento e as TAGS RFIDs podem facilmente ser utilizadas por outros usuários. Esse sistema pode ser concebido facilmente em ambientes empresariais, salas de reuniões e repartições públicas e não exige a necessidade de termos o sistema conectado a Internet. Por intermédio das avaliações realizadas observamos um tempo de resposta aceitável e uma maior robustez no controle de acesso, mesmo usando técnicas simples de reconhecimento de faces.*

1. Introdução

Uma cidade inteligente é um sistema urbano que utiliza tecnologia de informação e comunicação para trazer mais iteratividades tanto aos aspectos de infraestrutura como aos serviços públicos em geral. Essa iteratividade visa a acessibilidade e eficiência sob o ponto de vista dos cidadãos. Ademais, em uma cidade inteligente é de se esperar que a mesma esteja comprometida com o meio ambiente e com os elementos históricos e culturais. Nesse cenário a infraestrutura pode ser equipada com as mais avançadas soluções tecnológicas com o intuito de facilitar a interação do cidadão com os elementos urbanos e os diversos ambientes que eles estão inseridos [Pellicer et al. 2013].

Numa cidade inteligente é possível observar diversas possibilidades que as novas tecnologias oferecem para o fortalecimento do sistema urbano em geral. Como exemplo podemos citar: o monitoramento de poluentes tanto do ar como de rios e monitoramento e alarme sobre as condições climáticas de centros urbanos [Han et al. 2013]; as

diferentes soluções para a problemática de economia dos recursos energéticos em grandes cidades, por intermédio de *smart grids* [Weixiao et al. 2014]; a integração de veículos "inteligentes" capazes de interagir entre si para compartilhar informações de acidentes ou congestionamentos [Dias et al. 2014]; e a utilização de sensores inteligentes e sistemas embarcados com comunicação sem fio para automação de ambientes controlados, como prédios empresariais ou repartições publicas, permitindo assim a concepção de ambientes inteligentes [Sadri 2011]. Apenas delimitando o foco de nossos esforços, neste artigo, estaremos contribuindo apenas nesse último cenário apresentado.

Neste trabalho apresentamos um sistema embarcado para o controle de acesso em áreas restritas por intermédio da integração de reconhecimento facial e autenticação com RFID (*Radio Frenquency Identification-Identificação por Rádio frequência*) [Weis 2016]. Esse sistema visa aumentar a robustez no controle de acesso de prédios inteligentes.

A escolha da autenticação com RFID vem da sua grande utilização em diferentes aplicações, como por exemplo, na industria, em prédios e em lojas comerciais, tornando essa tecnologia objeto das mais diversas aplicações.RFID é um sistema sem contato, com o sistema de reconhecimento facial. Combinando o reconhecimento facial com o sistema de RFID irá adicionar mais robustez para qualquer sistema de segurança [Affandi et al. 2013].Existem diferentes tecnologias de RFID que consideram uma comunicação de longo, médio ou curto alcance. Na concepção desse trabalho utilizamos um leitor de médio alcance.

A etapa de reconhecimento facial é implementada usando técnicas leves de visão computacional que podem facilmente ser embarcadas em, por exemplo e no nosso caso, uma placa Beaglebone Black [1]. Para a leitura, controle e tratamentos dos dados dos sensores RFID e para decidir sobre a permissão ou não do acesso aos ambientes utilizamos uma placa Arduino UNO [2]. O Arduino UNO é uma plataforma de prototipagem eletrônica de hardware livre e possui baixo custo. Utiliza um microcontrolador da Atmel, com entradas/saídas digitais e analógicas, possui uma linguagem de programação semelhante ao C/C++ e foi projetado para facilitar a criação de protótipos.

As principais contribuições desse trabalho são: i. avaliação de técnicas de visão computacional convencionais em sistemas embarcados; ii. integração de duas soluções para controle de acesso num mesmo arcabouço; e iii. aumento na robustez dos sistemas de controle de acesso em ambientes inteligentes. Vale destacar que não é do escopo desse trabalho a proposição de novas técnicas de visão computacional para tais aplicações. Especificamente para as técnicas de visão computacional, estamos apenas interessados em avaliar a viabilidade da utilização dessas técnicas em ambientes embarcados voltados para a aplicação de controle de acesso em prédios inteligentes. Esse último ponto é importante para que, dado um cenário específico, o projetista possa avaliar e, consequentemente, identificar rapidamente quais as melhores técnicas que podem ser utilizadas. Por intermédio das avaliações realizadas observados um tempo de resposta aceitável e uma maior robustez no controle de acesso, mesmo usando técnicas simples de reconhecimento de faces.

Este artigo está organizado como segue: Seção 2 apresenta o embasamento teórico

[1]https://beagleboard.org/
[2]https://www.arduino.cc/

e os trabalhos correlatos; Seção 3 mostra o funcionamento do sistema proposto para o reconhecimento de faces; Seção 4 apresenta os principais resultados; e por fim, Seção 5 conclui o trabalho.

2. Embasamento teórico e trabalhos relacionados

Para o reconhecimento facial, o método mais utilizado na literatura é o *Eigenfaces* ou autofaces [Turk and Pentland 1991]. Este método retorna um conjunto de vetores e seu fundamento básico é a utilização de vetores de distribuição probabilística para a geração de dados da face. Ele utiliza PCA (*Principal Componente Analysis*) [jun Hu et al. 2015] para projetar e computar um subespaço para reconhecimento facial, o espaço de faces, através do treinamento de uma base de imagens, transformando as informações visuais em vetores. Esse espaço é definido pelos vetores autofaces que consistem na combinação linear dos pontos mais relevantes das imagens faciais originais, os quais são os autovetores da matriz covariância correspondente às imagens faciais originais. A Figura 1 mostra exemplos de autofaces.

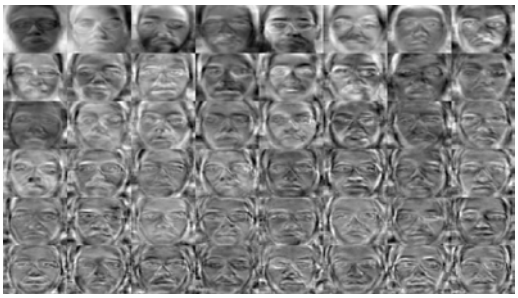


Figura 1. Conjunto de autofaces – imagens de faces após aplicar o método *Eigenfaces*.

Para compreendermos matematicamente o algoritmo de *Eigenfaces*, seja um conjunto de faces de treinamento $T = (T_1, T_2, \ldots, T_M)$. Seja M a quantidade de faces presentes no conjunto T, e m a face média (um vetor que representa a face média da base de imagens de faces),

$$m = \frac{1}{M} \sum_{i=1}^{M} T_i. \tag{1}$$

Seja $\mathbf{\Phi}_i = T_i - m$ a subtração da face média de cada imagem T_i, e a matriz $A = [\mathbf{\Phi}_1\, \mathbf{\Phi}_2\, \ldots\, \mathbf{\Phi}_n]$ onde cada coluna representa uma imagem $\mathbf{\Phi}_i$. A partir disso calculamos a matriz de covariância $C = A\, A^T$. Como a tarefa de encontrar autovalores e autovetores para a matriz C é custosa computacionalmente, então reduzimos a dimensão da matriz utilizando $C' = A^T\, A$. Utiliza-se a matriz de transformação $U = \mathbf{\Phi}\, C'$ para encontrar os autovetores e autovalores de C'. Em seguida, para projetar cada face no espaço das faces, aplica-se $\Omega = U^T\, \mathbf{\Phi}$.

Com o objetivo de reconhecer uma face de teste T_1 do banco de imagens, calcula-se a subtração da face média da face de teste $\mathbf{\Phi}_1 = T_1 - m$, para então projetar a face de teste no espaço das faces $\Omega = U^T\, \mathbf{\Phi}_1$. Por fim, calcula-se a distância euclidiana $\varepsilon_i = \|\Omega_1 - \Omega_i\|^2$ para cada face conhecida no banco de imagens, com $i = 1, 2, \ldots, M$.

A face do banco de imagens que possuir menor distância euclidiana com a face de teste permite fazer a conclusão de que são imagens bastante semelhantes ou iguais.

Outro método bastante utilizado para o reconhecimento facial é o *Haar Cascade* [Viola and Jones 2004]. Esse método é definido como uma estrutura contendo um encadeamento de classificadores do mais genérico ao mais específico [dos Reis 2014] e é baseado em aprendizagem de máquina, pois se utiliza de características previamente extraídas de um objeto para sua detecção. A função é treinada a partir de imagens positivas (imagens de faces) e imagens negativas (imagens sem faces). Utiliza retângulos como *features* (características) para localizar faces, de forma que possuam um valor único obtido subtraindo a soma dos *pixels* sob a região branca retangular da soma dos *pixels* sob a região preta retangular. Diferentes *features* podem representar um mesmo espaço na imagem, então são treinados diversos classificadores considerados fracos e depois combinados para gerar classificadores fortes, esses classificadores serão usados em sua cascata, a qual efetua uma busca por negativos. Ao serem encontrados, eles são descartados. Caso passe por todos os níveis da cascata, a região analisada possui uma face. A Figura 2 mostra alguns exemplos de *features*.

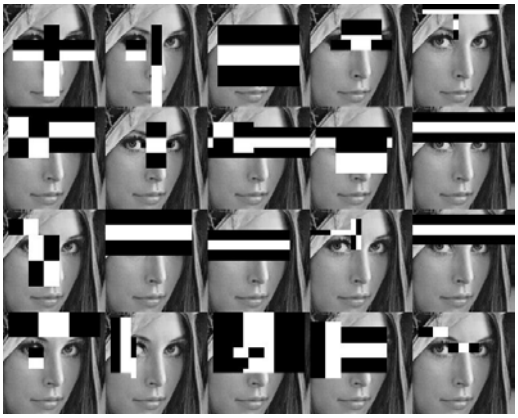


Figura 2. Modelos de *features* do *Haar Cascade*

As partes pretas e brancas das *features* representam a diferença de intensidade das cores entre as regiões da imagem, por exemplo: regiões mais escuras são representadas pela parte preta e as mais claras pela parte branca. Para o cálculo do *Haar Cascade*, precisa-se de uma imagem integral, isto é, uma matriz de mesma dimensão da imagem de entrada, onde (x, y) armazena a soma de todos os *pixels* do retângulo entre $(0, 0)$ e (x, y). Sendo $a(x, y)$ a imagem de entrada, a imagem integral $a'(x, y)$ é calculada através da seguinte recorrência:

$$S(x, y) = S(x, y - 1) + a(x, y)$$

e

$$a'(x, y) = a'(x - 1, y) + S(x, y),$$

onde $S(x, y)$ é a soma acumulada na linha, $S(x, -1) = 0$ e $a'(-1, y) = 0$. Após o cálculo da imagem integral, o valor relativo a cada retângulo da *feature* pode ser obtido através de quatro acessos a memória, isto é, quatro pontos (x, y) de *pixels*. Suponhamos uma subregião retangular D da *feature*: ponto (x_1, y_1) superior à esquerda; ponto (x_2, y_2)

superior à direita; ponto (x_3, y_3) inferior à esquerda e ponto (x_4, y_4) inferior à direita. Calculamos essa região como segue:

$$D = a'(x_4, y_4) + a'(x_1, y_1) - (a'(x_2, y_2) + a'(x_3, y_3)).$$

Para utilizar as técnicas de visão computacional, tratamento e processamento das imagem da face, utilizamos de forma adaptada para o nosso sistema embarcado o OpenCV (*Open Source Computer Vision library*) [3] que é um biblioteca que possui módulos de processamento de imagem e vídeo que contemplam as técnicas acima mencionadas. Além disso, o OpenCV possui suporte para as linguagem C/C++, JAVA, PYTHON e *Visual Basic*, o que facilita a portabilidade e difusão das soluções apresentadas.

Dentre os diversos trabalhos relacionados [Kail et al. 2007] e os citados no restante do trabalho sobre reconhecimento de faces temos o estudo apresentado por Hu et al. [jun Hu et al. 2015] que compararam dois métodos, *Principal Component Analysis* (PCA) e *Two-Dimensional Principal Component Analysis* (2DPCA), utilizando quatro base de dados de faces famosas com o objetivo de descobrir qual apresentava melhores resultados. Outro trabalho interessante é o apresentado por Kamencay et al. [Kamencay et al. 2014] que apresenta uma metodologia utilizando um algoritmo de *Principal Component Analysis* (PCA) combinado com *Canonical Correlation Analysis* (CCA) para aprender o mapeamento entre uma imagem facial 2D e dados de face 3D.

3. Sistema para controle de acesso

O sistema aqui proposto considera o controle de acesso em áreas restritas em, por exemplo, prédios públicos, laboratórios, arquivos e até salas de reuniões. Em suma o controle de acesso aqui proposto, consiste da utilização integrada de uma placa Beaglebone, para o reconhecimento de faces, e uma placa Arduino, para a autenticação via RFID, decisão da abertura da fechadura magnética e interação com o indivíduo via display.

O sistema inicia quando o acesso ao ambiente é solicitado, ou seja, quando à câmera identificar um indivíduo se aproximando da sala. Em seguida a própria câmera, por intermédio do microcontrolador embarcado nela (Beaglebone), aciona o módulo de reconhecimento facial. Ao mesmo tempo, o individuo pode autenticar sua chave RFID por intermédio de um chaveiro. Assim que a face é reconhecida e a chave RFID autenticada, a porta é aberta e um registro de acesso é armazenado, informando a entrada do indivíduo e o horário em que o mesmo entrou. Em caso de acesso negado, o sistema informa, via *display*, que o acesso foi negado. O gerente do sistema possui uma interface de acesso para que ele insira novos usuários, ou configure acessos temporários e programados.

As duas autenticações ocorrem de forma paralela, uma vez que temos um microcontrolador para o reconhecimento facial e outro para a autenticação da TAG RFID. O sistema de autorização de acesso tem os seguintes passos: i. a câmera captura uma imagem digital, onde é efetuada uma busca por uma região onde possa ter uma face; ii. a região é selecionada e passa por um processo de extração de dados relevantes; iii. os dados extraídos serão comparados com uma base de dados a fim de fazer o reconhecimento; iv. retorna para a placa Arduino o resultado se a face é conhecida; v. o leitor RFID coleta os dados do chaveiro; vi. é verificado se a face reconhecida corresponde a chave RFID

[3] http://opencv.org/

lida; e vii. se o usuário estiver cadastrado a placa Arduino abre a fechadura magnética, senão mostra uma mensagem de acesso negado no *display*.

A detecção da face, realizada pela Beaglebone, é obtida pelo método *Haar Cascade*. Após a detecção, a área da imagem que contém a face é normalizada usando funções disponibilizadas pelo OpenCV [4], isto é, a face é alinhada e são removidas imperfeições causadas pelo meio que as imagens são adquiridas. Após normalizar as imagens, é então utilizada a técnica *Eigenface*. O objetivo da aplicação desta técnica é extrair e codificar as características relevantes para reconhecimento das faces. Com esses dados é possível cadastrar um novo indivíduo no sistema ou realizar o reconhecimento utilizando como comparativo informações presentes na base de imagens do sistema.

Para conceber o sistema proposto, utilizamos uma placa Arduino UNO para receber e processar todos os dados e tomar a decisão de acesso a área restrita. O Arduino UNO possui um micro-controlador baseado no ATmega328, possuindo 14 pinos de entradas e/ou saídas digitais, das quais 6 podem ser usadas como PWM, um cristal oscilador de 16MHz, uma conexão USB, 6 entradas analógicas, uma entrada de alimentação e uma conexão ICSP. Para a montagem do protótipo Arduino, temos: um leitor RFID, um *display* LCD 2X16, um trafo 220Vac 12Vac, um relé 5V, um fecho magnético e um potênciomentro 10k. O *display*, relé e o leitor de RFID são ligados as portas de entrada e saída do Arduino. O usuário utiliza um chaveiro RFID de médio alcance que permite que a TAG seja lida através de barreias e objetos, não necessitando o contato direto. A Figure 3 mostra o protótipo Arduino desenvolvido.

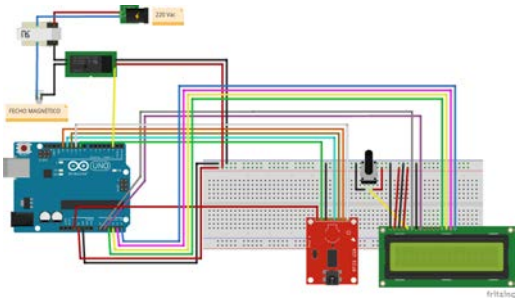


Figura 3. Protótipo do controle de acesso

Nesse protótipo, quando uma TAG é lida pelo leitor RFID o Arduino interpreta seus dados e verifica se a TAG está habilitada para acessar o ambiente, se estiver e a face foi reconhecida o Arduino aciona a porta que está ligado o relé, fazendo com quer o fecho magnético receba um pulso elétrico e destrave a porta, assim dando acesso ao usuário.

A comunicação entre a Beaglebone e o Arduino é feita pela porta serial. O Arduino cria uma comunicação serial com o microcontrolador e através dessa porta serial o sistema envia a informação do reconhecimento da face. Essa comunicação pode ser feita de outras maneiras, por exemplo, via bluetooth ou transmissor de radio frequência. Como a Beaglebone possui um sistemas operacionais embarcado optamos por usar a comunicação serial, pois o tempo de envio e recebimento de informação é mais rápido. Ademais, ao utilizar a Beaglebone com o modulo de radio frequência algumas funções,

[4]http://opencv.org/

como a saída de vídeo, é desabilitada o que dificulta o seu uso integrado com uma câmera para o reconhecimento de face.

Para o processamento das imagens, utilizamos a Beaglebone Black que possui um processador ARM Cortex A8 1GHz, com 512M de memória RAM, saída de vídeo HDMI, 2GB de eMMC de memória flash, Kernel do Linux 3.8 e sistema operacional Debian 7.8.

Acoplada a Beaglebone realizamos testes com duas câmeras. A primeira delas, bem sofisticada, é a D-link – HD Wireless N Cube Network Camera – capaz de capturar imagens em alta resolução 1 megapixel HD-720P garantido a captura dos detalhes que necessitamos, usa uma compressão de vídeo H.264, MPEG-4 e MJPEG. A câmera possui a seguinte especificação: 1/4 1 megapixel progressiva CMOS sensores, iluminação mínima de 1.0 lux, zoom digital 10x, comprimento fixo 3.45 mm, abertura de F2.0, ângulo de visão de H 57,8º, V 37,8º, D 66º. A outra câmera, uma simples web Cam, é a Goldship hd, capaz de gerar imagem com qualidade HD e que possui a seguinte especificação: conexão com o computador via USB, classe de vídeo USB (UVC), resolução Máxima em HD 720p e taxa de quadros acima de 30fps.

4. Resultados e discussões

Para identificar o impacto do uso do reconhecimento facial de forma embarcada e considerando, em especial, o tempo de processamento das técnicas utilizadas realizamos experimentos com 10 faces, 5 cadastradas na base de imagens e 5 não cadastradas. A base continha 12 faces diferentes cadastradas: 100 fotos foram tiradas de cada face para efetuar o cadastro, resultando em um total de 1.200 imagens.Ressaltando que foram testes iniciais, nos trabalhos futuros será incluída um base de dados maior e os testes serão realizados com mais faces.

Os testes visam o desempenho da placa embarcada na detecção da face e no seu reconhecimento. Salientando que os testes não são para analisar o desempenho dos algoritmos de detecção e reconhecimento da face e, sim a placa embarcada. Buscar por algoritmo que possuam o melhor desempenho qualitativo esta fora do escopo desse trabalho.

Os testes foram efetuados em tempo real utilizando a câmera D-Link com e resolução 352×288. As faces utilizadas nos testes foram postas em 3 posições: frontal, alta e lateral. A posição frontal refere-se à câmera posicionada de frente à face; na posição alta a câmera é posta em um local acima da face, porém o indivíduo deve olhar para a câmera (simula-se uma câmera na parede em nível mais alto que o indivíduo); e a posição lateral é semelhante à posição alta, mas forma um ângulo diferente de 90 graus com a face, escondendo parcialmente um dos lados da face. Para cada posição foram efetuadas 30 capturas de imagens e repetido o processo para 3 graus de confiança diferentes, 0.975, 0.980 e 0.985, resultando em 300 análises para cada grau, com um total de 900 análises.

A Tabela 1 exibe o resultado com faces existentes no banco de imagens. Configurando o nível de confiança em 0.975, foram obtidos resultados muito bons para a posição frontal e alta, basta verificar a taxa de acerto, 94% e 74%, respectivamente. Já na posição lateral o reconhecimento correto ficou abaixo de 50% e mais próximo dos resultados errados e em 15.3% das imagens não foi obtido êxito na detecção da face.

Tabela 1. Indivíduos presentes na base.

Confiança	0.975	0.980	0.985	
Frontal	0.940	0.794	0.173	Certo
	0.060	0.180	0.053	Errado
	0	0.026	0.774	Desconhecido
Alta	0.740	0.507	0	Certo
	0.253	0	0	Errado
	0.007	0.493	1	Desconhecido
Lateral	0.467	0.300	0	Certo
	0.380	0.100	0	Errado
	0	0.520	0.967	Desconhecido

Tabela 2. Indivíduos não presentes na base.

Confiança	0.975	0.980	0.985	
Frontal	0	0.613	1	Certo
	1	0.387	0	Errado
	0	0	0	Desconhecido
Alta	0.060	0.947	0.987	Certo
	0.840	0,007	0	Errado
	0	0	0	Desconhecido
Lateral	0.007	0.920	0.980	Certo
	0.867	0	0	Errado
	0	0	0	Desconhecido

Aumentando o nível de confiança para 0.980 os resultados se mantiveram consistentes. Na posição alta, mesmo com reconhecimento correto um pouco acima de 50%, os resultados errados foram zerados, melhorando a eficiência do algoritmo. Com o nível de confiança em 0.985, só houve reconhecimento correto para a posição frontal; contudo, com taxa de acerto baixa e recebendo como saída muitas vezes "face desconhecida".

A Tabela 2 mostra o resultado dos testes efetuados com faces não cadastradas. Para o nível de confiança 0.975, as saídas do reconhecimento facial foram em sua grande maioria errados; contudo, ao aumentar o nível de confiança, o retorno do algoritmo foi também em sua grande maioria correto.

Nas duas tabelas citadas foi observado que o resultado bom na primeira refletia em um resultado ruim na segunda. Por esse motivo a Tabela 3 foi criada, unificando os resultados das tabelas em uma só. Os valores para o nível de confiança 0.980 para a posição frontal e alta de forma geral ficaram bastante satisfatórios, alcançando um bom índice de acerto e baixos índices de erros. O resultado para a posição lateral ficou com uma boa porcentagem de acerto, o motivo é a alta taxa de acerto das faces não cadastradas, embora a taxa de acerto para faces cadastradas seja baixo.

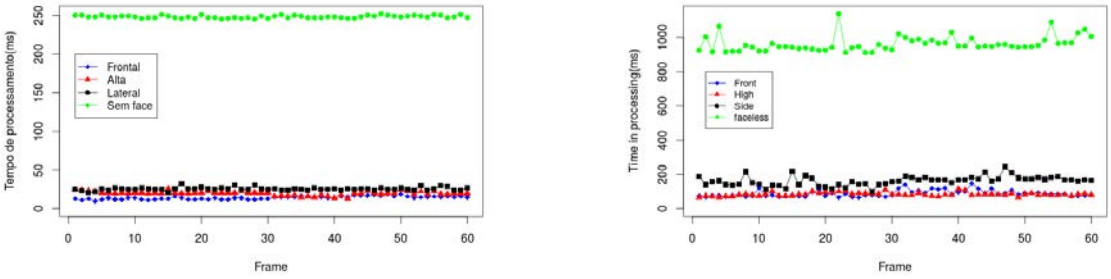
Tabela 3. Indivíduos presentes e não presentes na base.

Confiança	0.975	0.980	0.985	
Frontal	0.470	0.703	0.587	Certo
	0.530	0.283	0.027	Errado
	0	0.013	0.386	Desconhecido
Alta	0.400	0.727	0.494	Certo
	0.547	0.003	0	Errado
	0.003	0.247	0.500	Desconhecido
Lateral	0.237	0.610	0.490	Certo
	0.623	0.050	0	Errado
	0	0.260	0.483	Desconhecido

Para comparar o tempo de resposta da placa, executamos em tempo real o módulo de reconhecimento facial também em um computador. O computador utilizado possui um sistema operacional linux ubuntu 14.04.LTS, memória RAM de 8G, processador intel Core i7-3770 CPU@3.40GHz x 8, placa gráfica AMD Radeon HD 6450 e disco rígido de 1T.

Os gráficos da Figura 4 ilustram o tempo de execução em ambos os casos, vale destacar que, com o objetivo melhorar a visualização, as escalas das figuras no eixo-y não são as mesmas. Pode-se ver que apenas quando o sistema esta executando sem ter nenhuma face a frente da câmera é que o processamento é muito alto. No entanto, nos

outros cenários o tempo de processamento ao se utilizar a placa se assemelha a de um servidor. Um fato importante é que estamos considerando a execução isolada de apenas uma requisição. No caso de termos várias requisições, certamente um maior poder computacional será requerido para o processamento no servidor. Vale salientar que nossa proposta é que o sistema seja desconectado a Internet, logo para utilizarmos um PC teríamos que ter uma máquina exclusiva para isso.



(a) Computador *desktop* (b) Placa

Figura 4. Tempo de resposta referente ao reconhecimento facial.

Uma vez que foi possível identificar o baixo impacto do uso do reconhecimento facial de forma embarcada realizamos testes para identificar a robustez a validade de nosso sistema completo, ou seja, considerando o reconhecimento de face e a autenticação via RFID simultaneamente.

Foram realizados testes com quatro pessoas, das quais três possuíam suas faces cadastradas no sistema de reconhecimento facial e uma não possuía, foi utilizado dois chaveiros RFIDs, um com acesso permitido e outro sem acesso.

Foi realizado quatro tipos de teste: i. faces cadastradas e o chaveiro RFID valido, nesse teste o sistema funcionou corretamente para todas as tentativas reconhecendo a face e validando com o chaveiro RFID assim liberando o acesso do usuário ao ambiente; ii. as faces cadastradas e o chaveiro RFID sem acesso permitido ao ambiente, neste caso sistema reconheceu as faces mas não liberou o acesso, pois o chaveiro não validou o acesso; iii. com a face não cadastrada e o chaveiro RFID valido, neste caso o sistema algumas vezes a face, gerando um falso positivo então o sistema liberava o acesso; iv. face não cadastrada e o chaveiro RFID sem acesso permitido, neste teste mesmo gerando algum falso positivo o acesso ao ambiente era negado pois a etiqueta não era valida para acessar o ambiente. Para a detecção e reconhecimento facial, foram utilizadas as técnicas *Eigenfaces* e*Haar Cascade* descritas na sessão 2 deste trabalho.

Os teste foram feitos tanto no PC como na placa, em ambos os casos, foram observados os mesmos resultados para os testes, só que com um tempo maior para fazer o reconhecimento facial na placa, pois o tempo de processamento e acesso a memória são superiores dos que observados no computador *desktop*.

5. Conclusão e trabalhos futuros

Este artigo apresentou uma proposta de um sistema embarcado o controle de acesso considerando o reconhecimento de faces e autenticação via RFID. O estudo aqui realizado

mostra a viabilidade do sistema embarcado, sem a dependência de uma central. Como o sistema é executado na própria câmera, sem acesso a Internet, aumentando a segurança do sistema. Mesmo com um atraso na detecção, quando comparado com uma execução num computador *desktop*, o sistema funcionou como esperando só liberando o acesso para pessoas autorizadas.

Para trabalhos futuros, pretendemos fazer o rastreamento de pessoas dentro do prédio. Planejamos inserir outros elementos de sensoriamento ao sistema, no caso, sensores de presença e temperatura, sensor de luminosidade e controlador das lâmpadas e ar-condicionado, para assim realizarmos o monitoramento completo do sala ou área considerada. Pretendemos também usar uma rede e protocolo de comunicação para que a base de dados de reconhecimento seja compartilhada entre várias áreas do prédio.

Referências

Affandi, A., Awedh, M., Husain, M., and Alghamdi, A. (November 2013). Rfid and face recognition based security and access control system. *International Journal of Innovative Research in Science, Enginnering and Technology*.

Dias, J. A. F. F., Rodrigues, J. J. P. C., and Zhou, L. (2014). Cooperation advances on vehicular communications: A survey. *Vehicular Communications*, 1(1):22 – 32.

dos Reis, W. A. R. (2014). Detecção de sinais de trânsito através do método de classificação adaboost. *UNOPAR Científica Ciências Exatas e Tecnológicas*, 12(1).

Han, G., Shu, L., Pathan, A. S. K., Rodrigues, J. J. P. C., and Mellouk, A. (2013). Wireless sensor networks based on environmental energy harvesting. *The International Journal of Distributed Sensor Networks*, 2013(816063):2.

jun Hu, J., zheng Tan, G., 0gang Luan, F., and Libda, A. S. M. L. (2015). 2DPCA versus PCA for face recognition. *Journal of Central South University*, 22(5):1809–1816.

Kail, K., Williams, C., and Kail, R. (2007). Access control system with rfid and biometric facial recognition. US Patent App. 11/790,385.

Kamencay, P., Hudec, R., Benco, M., and Zachariasova, M. (2014). 2D-3D face recognition method based on a modified ccapca algorithm. *Int J Adv Robot Syst*, 11:36.

Pellicer, S., Santa, G., Bleda, A. L., Maestre, R., Jara, A. J., and Skarmeta, A. G. (2013). A global perspective of smart cities: A survey. In *7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*.

Sadri, F. (2011). Ambient intelligence: A survey. *ACM Computing Surveys*, 43(4):36:1 – 36:66.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.

Weis, S. A. (2016). Rfid (radio frequency identification): Principles and applications.

Weixiao, M., Ma, R., and Chen, H. H. (2014). Smart grid neighborhood area networks: A survey. *IEEE Network*, 28(1):24 – 32.