

IMPLEMENTAÇÃO DE UM SISTEMA DE RECONHECIMENTO DE FACES UTILIZANDO RASPBERRY PI

Ed' Wilson Tavares Ferreira¹

Mattheus Nunes Araújo²

Luiz Felipe Silva³

Valtemir Emerencio Nascimento⁴

Ruy de Oliveira⁵

RESUMO: Diversos pesquisadores têm demonstrado interesse em projetos baseados em reconhecimento e classificação de padrões, sobretudo os aplicados em reconhecimento de faces, visto a diversidade de cenários em que esta tecnologia pode ser utilizada. Neste artigo é apresentado o resultado da implementação do sistema de reconhecimento de faces baseado em placas Raspberry Pi, com o emprego do software *Wolfram Mathematica*. Os testes foram realizados a partir da criação de um conjunto de dados composto por fotografias de 10 pessoas. O sistema foi desenvolvido com diversos dispositivos (*buzzer*, LED's, display LCD, botões e RaspiCam), conectados a GPIO (*General Purpose Input/Output*) do Raspberry Pi. O software foi desenvolvido com a implementação dos seguintes métodos de classificação e reconhecimento de padrões: regressão logística (*logistic regression*), *Markov*, *random forest*, interpolação por vizinho mais próximo (*nearest neighbor*), *naive Bayes*, redes neurais (*neural networks*) e máquina de vetores suporte (*support vector machine*) e bons resultados foram obtidos, com acurácia de 90% de acerto, que demonstra que a estratégia adotada é bastante promissora.

PALAVRAS-CHAVE: Reconhecimento facial, Raspberry Pi e Python.

ABSTRACT: Several researchers have shown interest in projects based on recognition and pattern classification, especially applied to face recognition, mainly the diversity of scenarios in which this technology can be used. This paper presents the results of implementation of face recognition system based on Raspberry Pi boards with the use of Wolfram Mathematica software. The tests were performed with the data set consisting of 10 photos of persons. The system was developed with several devices (*buzzer*, LED, LCD display, buttons and RaspiCam),

1 Doutor em Engenharia Elétrica e Professor do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso - *Campus* Cuiabá. *E-mail:* edwilson.ferreira@ifmt.edu.br

2 Discente de Engenharia da Computação pelo Instituto Federal de Educação, Ciência e Tecnologia do Mato Grosso - *Campus* Cuiabá. *Email:* mattheus_nunes@hotmail.com

3 Discente de Engenharia da Computação pelo Instituto Federal de Educação, Ciência e Tecnologia do Mato Grosso - *Campus* Cuiabá. *Email:* luizfelpesilva91@gmail.com

4 Doutor em Engenharia Elétrica e Professor do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso - *Campus* Cuiabá. *E-mail:* valtemir.nascimento@cba.ifmt.edu.br

5 Doutor em Engenharia Elétrica e Professor do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso - *Campus* Cuiabá. *E-mail:* ruy.oliveira@cba.ifmt.edu.br

connected to the Raspberry Pi GPIO. The software was developed with the implementation of the follow-up methods of classification and pattern recognition: logistic regression, Markov, random forest, nearest neighbor, naive Bayes, neural networks and machine support vectors machine and good results were obtained, showing that the adopted strategy is very promising.

KEYWORDS: Facial recognition, Raspberry Pi and Python.

INTRODUÇÃO

O reconhecimento de faces é uma metodologia responsável pela identificação de um indivíduo, baseada em suas características faciais. As pesquisas nessa área têm apresentado avanço significativo e, atualmente, o reconhecimento facial é utilizado em diversas áreas: desde entretenimento (videogames) até segurança nacional (passaportes e vídeo vigilância). Recentemente, foi noticiado na imprensa que engenheiros chineses, em uma parceria entre a Universidade Tsinghua (Pequim) e a empresa de segurança Tzekwan Technology, desenvolveram um protótipo de caixa eletrônico com reconhecimento de faces, visando diminuir os crimes envolvendo caixas eletrônicos (INFO ABRIL, 2015). Em Uberlândia - MG, foi instalado, nos ônibus, um sistema de reconhecimento de faces, numa tentativa de redução de fraude (FOLHA DE SÃO PAULO, 2014). Além disso, em Anápolis - GO, a Escola Municipal Anapolino de Faria implantou um sistema de reconhecimento de faces para controlar a frequência dos alunos. Esse sistema realiza o reconhecimento do rosto dos estudantes e registra a presença no sistema conectado à internet e, no caso da ausência, os pais são notificados via SMS (G1 GLOBO, 2015).

Um dos principais problemas no reconhecimento de faces, no momento, refere-se ao aprimoramento dos algoritmos e técnicas empregadas no reconhecimento de faces, principalmente quando há variação no ambiente, nas posições das faces e nas expressões faciais, além da presença de acessórios (como os óculos, lenços e outros objetos).

No projeto ora apresentado, foi proposto o estudo e a implementação de um sistema de reconhecimento facial para ser executado na plataforma computacional *Raspberry Pi*, como um projeto de iniciação científica. O *Raspberry Pi* possui baixo custo em comparação com computadores convencionais (desktops e notebooks) e possui dimensões compactas, ideal para mobilidade. Além disso, o seu sistema operacional (*Raspbian OS*) suporta o software *Wolfram Mathematica* que, por sua vez, dispõe de uma vasta biblioteca de funções que implementam algoritmos de reconhecimento facial.

Uma contribuição importante, como resultado desta pesquisa, foi a implementação do sistema de reconhecimento facial com o emprego de uma plataforma computacional de baixo custo integrando o software *Wolfram Mathematica*, juntamente com o Python. O sistema implementado pode ser utilizado para melhorar a segurança, com o acionamento de catracas e portas com fechaduras eletrônicas. Outra contribuição foi a criação de um banco de imagens com alunos dos cursos superiores do Instituto Federal de Mato Grosso – IFMT. Enquanto outras pesquisas utilizam um conjunto de dados prontos para a sua avaliação, esta pesquisa, ao utilizar um conjunto próprio, tem-se, para a avaliação, um cenário heterogêneo. O conjunto

de fotos geradas nesta pesquisa apresenta pessoas com diversos tipos de vestimentas e acessórios (lenços e óculos), além de expressões faciais diferentes. Estas características contribuíram para disponibilizar um ambiente real de testes para sistema desenvolvido.

TRABALHOS RELACIONADOS

O reconhecimento de faces é uma das mais importantes aplicações de processamento de imagens. Atualmente, nota-se um crescimento significativo no interesse neste tipo de pesquisa em diversas áreas e aplicações. Em redes sociais, como o Facebook e o Google+, o reconhecimento está cada vez mais eficiente. Em várias situações, o software é capaz de reconhecer todas as pessoas em uma fotografia (desde que tenham perfil na rede). Tal automatização é complexa e nem sempre é possível o mesmo resultado quando comparado à capacidade do cérebro humano (PANDYA; RATHOD; JADAV, 2014).

No artigo intitulado “Face Recognition System On Raspberry Pi” (NIKISINS et al., 2015), os autores propuseram o desenvolvimento de um sistema embarcado de reconhecimento facial baseado na plataforma *Raspberry Pi*. O software foi escrito em C++ e utilizou a biblioteca OpenCV para aquisição e detecção da face, já o hardware foi composto pelo *Raspberry Pi*, por uma placa de controle, uma câmera USB e um display LCD. Para realizar o reconhecimento facial, os autores empregaram um método holístico denominado *Local Binary Pattern* (LBP), que foi aplicado às faces detectadas e as transformou em um histograma. Dessa forma, o reconhecimento é efetivado quando há compatibilidade entre o histograma gerado e o histograma armazenado. O método empregado para comparar os histogramas foi o de intersecção. Os autores aplicaram o sistema completo ao banco de dados FERET (*Facial Recognition Technology*) e obtiveram resultados promissores: taxa CMC (*Cumulative Match Characteristic*) de 99,33 % e taxa EER (*Equal Error Rate*) de 1 %. (NIKISINS et al., 2015). Contudo, os diferentes níveis de iluminação de um ambiente e as variações nas expressões faciais não foram fatores destacados no artigo.

No artigo “Reconhecimento de Faces Utilizando Análise de Componentes Principais e a Transformada Census” (RIBEIRO; CHIACHIA; MARANA, 2010), o fator variação luminosa foi analisado detalhadamente. Em contraste com o método holístico utilizado no trabalho anterior, os autores desse artigo empregaram um método estatístico denominado Análise de Componentes Principais (*Principal Component Analysis* – PCA), uma vez que esse método reduz a dimensionalidade da operação. Para estudar os efeitos das variações luminosas, os autores aplicaram o PCA a um conjunto de dados em duas situações: PCA puro e PCA associado com a Transformada Census, que é um filtro de imagens. O emprego da Transformada Census diminuiu a taxa de erros no reconhecimento de imagens que apresentavam diferenças de iluminação. (RIBEIRO

CHIACHIA; MARANA, 2010). Os autores não apresentam informações sobre a arquitetura tecnológica empregada na realização dos seus experimentos.

Observa-se, ainda, a presença do reconhecimento facial nos smartphones, empregado em funções de identificação do usuário e em jogos. Com a disponibilização da biblioteca OpenCV, o desenvolvimento deste tipo de aplicação tornou-se mais fácil e rápido, e bons resultados têm sido obtidos nesses dispositivos portáteis (SHEN; HU; YANG, 2014).

Finalmente, nota-se que novas abordagens de detecção estão surgindo. É o caso da técnica apresentada no trabalho “Face Recognition Using Neuro-Fuzzy Inference System” (MEHTA et al., 2014), composta por uma combinação de redes neurais com lógica *fuzzy*. Essa nova abordagem apresentou melhor acurácia do que técnicas clássicas, como a *EigenFace*, o modelo oculto de Markov e o reconhecimento de face baseado em atributos (MEHTA et al., 2014), porém os autores não informaram qual hardware e sistema operacional foram utilizados.

METODOLOGIA

Hardware e Software

Para a implementação do sistema proposto, foi utilizado o microcomputador *Raspberry Pi* modelo B+, como apresentado na Figura 1, que possui as seguintes especificações: dimensões de 85mm x 56mm, processador Broadcom SoC 700MHz, 512MB de memória RAM, conector de energia microUSB, saída HDMI, saída de áudio A/V, porta Ethernet, 4 portas USB, socket para cartão microSD e 40 pinos de entrada/saída, conhecidos como GPIO (*General Purpose Input/Output*) (ADAFRUIT, 2015). Além disso, permite o acoplamento de diversas interfaces, como o módulo de câmera (*RaspiCam*), que é capaz de efetuar a captura de imagens e vídeo.

O sistema também utilizou três LED's comuns de 5mm (amarelo, vermelho e verde), duas chaves tácteis, um *buzzer* (dispositivo emissor de som), um display LCD 16x2, resistores e *jumpers* diversos. Esses dispositivos foram conectados ao GPIO com o objetivo de atuarem como uma interface entre o usuário e o sistema.

O software *Wolfram Mathematica* foi concebido, inicialmente, para atuar como um software de álgebra computacional. Contudo, apresentou expressiva evolução nos últimos anos, tornando-se um ambiente computacional com aplicações em diversas áreas do conhecimento: engenharia, biologia, química, processamento de imagens, finanças e estatística. Funciona, também, como ambiente de desenvolvimento de sistemas, além de possuir editor de documentos e apresentações. É importante salientar que a sua licença de uso é gratuita na plataforma *Raspberry Pi*, contribuindo significativamente para o desenvolvimento de projetos e pesquisas (WOLFRAM, 2015), portanto foi empregado como parte do arcabouço de software para o sistema desenvolvido.

Figura 1. *Raspberry Pi* – Modelo B+.



Reconhecimento Facial

Definidos os componentes essenciais do hardware e software para a implementação do sistema, o passo seguinte foi a elaboração de um conjunto de dados composto por 10 fotografias de 10 pessoas, sendo que dessas 10 fotografias, 1 foi reservada para o posterior reconhecimento. Em seguida, a função *FindFaces* do software *Wolfram Mathematica* foi aplicada às amostras com o objetivo de identificar as faces. A aplicação dessa função resulta numa lista de coordenadas que representam a ROI (*Region of Interest*) (WOLFRAM, 2015). A identificação precisa dessa região foi necessária para que a função *ImageTrim* pudesse recortá-la, evidenciando apenas a face. Devido a uma limitação referente à arquitetura do microprocessador do *Raspberry Pi*, a etapa de identificação das faces foi executada em um *desktop* executando o sistema operacional Windows 7. Nas fotografias em que as pessoas estavam em posição perfil ou possuíam algum acessório, a identificação não foi possível. Diante desse cenário, o recorte das faces foi realizado manualmente.

A etapa seguinte foi a de aprendizado de máquina (*machine learning*), necessária para realizar o reconhecimento facial. Essa etapa caracterizou-se pela aplicação da função *Classify* ao conjunto de dados, associando cada fotografia (exemplo) ao indivíduo (classe) correspondente. A função *Classify* gera uma *ClassifierFunction* baseada nos exemplos e classes fornecidas como parâmetros, que posteriormente pode ser utilizada para classificar outros dados (WOLFRAM, 2015).

É possível especificar o método de classificação ao executar a função *Classify*. Os métodos disponíveis são: regressão logística (*logistic regression*), *Markov*, *random forest*, interpolação por vizinho mais próximo (*nearest neighbor*), *naive Bayes*, redes neurais (*neural networks*) e máquina de vetores suporte (*support vector machine*). Caso o

método não seja especificado (automático), o *Wolfram Mathematica* tentará escolher o método mais apropriado para aquela operação (WOLFRAM, 2015).

A regressão logística faz parte de uma categoria de modelos estatísticos que permite prever um resultado discreto, a partir de um conjunto de variáveis que podem ser contínuas, discretas, dicotômicas, ou uma mistura destas. Sendo assim, a variável dependente ou resposta é dicotômica, como a presença/ausência ou o sucesso/fracasso. A variável dependente na regressão logística é geralmente dicotômica, a variável dependente pode assumir o valor 1 com probabilidade de sucesso ou o valor 0 com probabilidade de falha. O objetivo da regressão logística é prever corretamente a categoria de resultado para casos individuais usando o modelo mais criterioso (CONNOR, 2015).

O método de *Markov* é um modelo estatístico que tem o desafio de determinar os parâmetros ocultos a partir dos parâmetros observáveis. Esse modelo é muito utilizado em aplicações de reconhecimento de padrões, tais como a fala, a escrita, os gestos e a bioinformática. Os processos de Markov são modelados por sistemas de transições de estados, os quais são representados em termos de seus vetores probabilísticos, que podem variar no espaço temporal (discreto ou contínuo), e as transições entre os estados são probabilísticas e dependem apenas do estado concorrente (FINK, 2014).

O método de *Naive Bayes* baseia-se no chamado teorema bayesiano, que consiste em classificar objetos a partir de estados anteriores. Dessa forma, após a chegada de um novo caso, decide-se a que rótulo de classe pertence com base nos objetos de saída. A análise bayesiana é conhecida como a propriedade anterior baseada nas experiências anteriores. É muito utilizada para a previsão de resultados (DELL, 2013).

O método *Nearest Neighbors* ou vizinho mais próximo consiste em encontrar um número pré-definido de amostras de treinamento mais próximos, em distância, para o novo ponto. O número de amostras pode ser uma constante ou variar com base na densidade local de pontos (PEDREGOSA, 2011).

As redes neurais são um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência obtida. Uma rede neural é composta por várias unidades de processamento, que são conectadas por canais de comunicação associados a determinados pesos. Assim, as unidades realizam operações apenas sobre seus dados locais, que são entradas recebidas por suas conexões (CARVALHO, 2005).

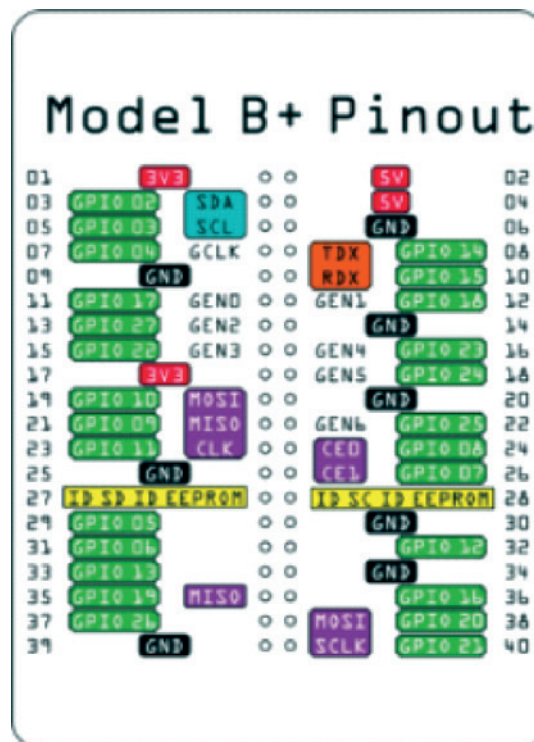
O *random forest* ou floresta aleatória é um método estimador que se encaixa em uma série de classificadores de árvore de decisão, baseados em várias subamostras do conjunto de dados. Utiliza-se uma média para melhorar a precisão da previsão e o controle sobre a montagem do conjunto de decisão (PEDREGOSA, 2011).

Ao aplicar-se a função *Classify* em modo automático ao conjunto de dados proposto, o resultado foi o método regressão logística como *ClassifierFunction*. Contudo, todos os métodos possíveis foram testados com a finalidade de comparar o desempenho de cada um, corroborando ou não com o método automático escolhido pelo software.

Interface do sistema

O *Raspberry Pi* modelo B+ possui 40 pinos de entrada e saída de dados, que são a interface física entre a placa e o ambiente externo. Esses pinos são denominados GPIO e podem ser utilizados em diversos cenários, desde o acendimento de um LED até a leitura dos dados de um sensor. Observa-se que há duas formas bem distintas de se referenciar os pinos do GPIO: endereçamento físico e endereçamento lógico, como mostrado na Figura 2. Esse fato se deve às funções especiais que alguns pinos podem desempenhar. Os pinos 08 e 10 (GPIO 15 e GPIO 16) podem realizar comunicação serial (TDX RDX) entre o *Raspberry Pi* e algum dispositivo compatível, por exemplo.

Figura 2. *Raspberry Pi* B+: GPIO.



O *Wolfram Mathematica* possui um conjunto de funções que permite o acesso direto aos dispositivos conectados ao computador. São algumas dessas funções: *DeviceConfigure*, *DeviceOpen*, *DeviceRead* e *DeviceWrite*. No caso específico do *Raspberry Pi*, o *Wolfram Mathematica* possui o dispositivo *RaspiCam* e o GPIO pré-configurados, o que facilita a comunicação com eles.

Para ler uma imagem do *RaspiCam*, basta executar o comando *DeviceRead*["*RaspiCam*"]. Além disso, para ler o valor de um pino, executa-se o comando *DeviceRead*["GPIO", *p*]. Similarmente, para escrever um valor em um pino, executa-se o comando *DeviceWrite*["GPIO", *p* -> *v*]. O valor *v* de um pino *p* é sempre 0 ou 1, e o endereçamento

utilizado pelo *Wolfram Mathematica* é o lógico, que será utilizado no decorrer desse trabalho também.

Os dispositivos externos propostos foram conectados ao GPIO com o auxílio de uma matriz de contato (*protoboard*) e diversos conectores (*jumpers*). Os LED's verde, vermelho e amarelo foram conectados aos pinos GPIO 04, GPIO 17 e GPIO 27, respectivamente. Os dois *pushbuttons* foram conectados aos pinos GPIO 09 e GPIO 10. Além disso, para conectar os LED's e os *pushbuttons*, foi necessário o uso de resistores de 330Ω. O *buzzer* foi conectado ao pino GPIO 22.

O display LCD 16x2, apresentado na Figura 3, possui 16 pinos e possui dois modos de operação: 4 bits e 8 bits. O modo de operação utilizado foi o de 4 bits, visto que dispensa a conexão de 4 pinos e apresenta desempenho semelhante ao modo de operação de 8 bits. Portanto, os pinos 7, 8, 9, e 10 não foram utilizados e os pinos 11, 12, 13 e 14 foram conectados aos pinos GPIO 13, GPIO 06, GPIO 05 e GPIO 11, respectivamente. O pino 4 (*Register Select*) é responsável por controlar o local de escrita de dados na memória do LCD e foi conectado ao pino GPIO 026. Já o pino 6 (*Enable*) apresenta a função de habilitar a escrita de dados e foi conectado ao pino GPIO 19. O pino 5 (*Read/Write*) seleciona o modo de leitura ou escrita. Uma vez que foi necessária apenas a escrita, esse pino foi aterrado, desabilitando o modo de leitura.

Em termos de interface, foi definida uma função específica para cada elemento proposto. O acendimento dos LED's sinaliza três situações básicas: o carregamento do banco de imagens (LED amarelo), acesso autorizado (LED verde) e acesso negado (LED vermelho). Similarmente, o display LCD é responsável por exibir mensagens correspondentes a essas situações: "Carregando", "Acesso Autorizado" e "Acesso Negado", além de exibir a mensagem "Pronto" para avisar que o carregamento do banco de imagens foi concluído e que o sistema está apto para prosseguir. O primeiro *pushbutton* é responsável pela entrada de dados, ativando o *RaspiCam* para obter uma imagem do usuário. Ao pressionar-se o segundo *pushbutton*, o sistema é encerrado. Ao passo que os LED's e o LCD atuam como sinalizadores visuais, o *buzzer* atua como sinalizador sonoro. A Figura 4 contém o protótipo desenvolvido e na Figura 5 é apresentada uma amostra do conjunto de dados das fotografias.

Figura 3. Pinagem do Display LCD 16x2.

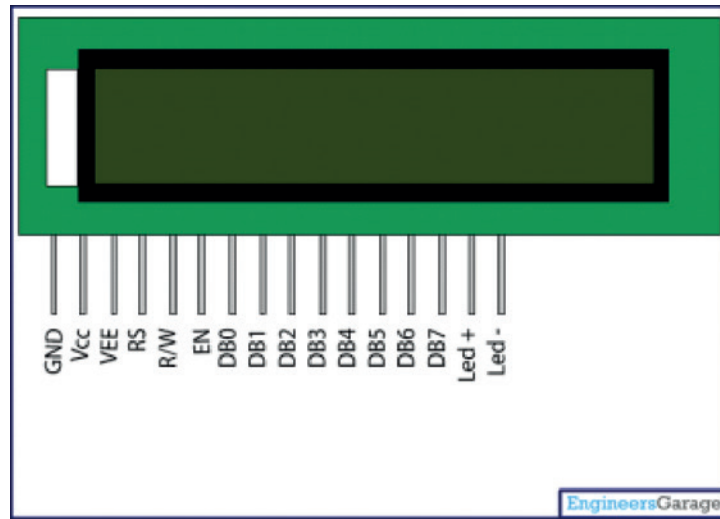


Figura 4. Hardware do Sistema.

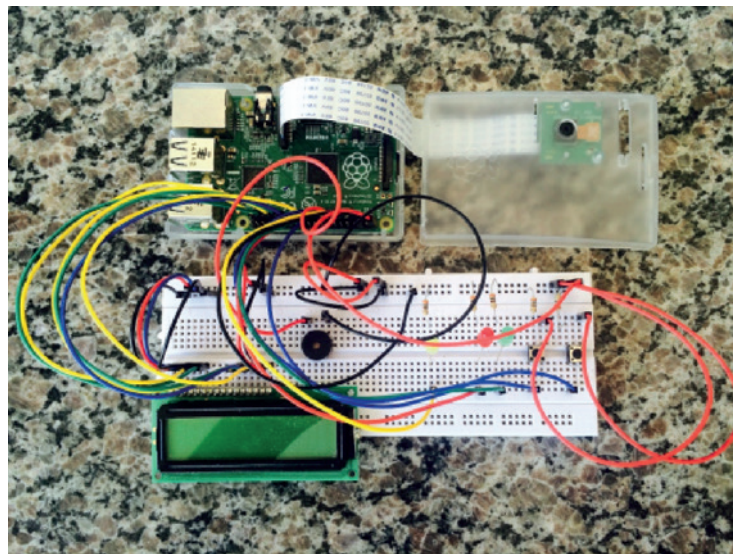


Figura 5. Conjunto de Fotos.



RESULTADOS E DISCUSSÃO

Para avaliar o desempenho e a eficiência do sistema proposto, foram realizados vários testes. O primeiro teste consistiu no cálculo do tempo decorrido para o treinamento (*machine learning*) do conjunto de dados, utilizando-se os cinco métodos disponíveis para a plataforma *Raspberry Pi*. O tempo decorrido foi calculado pela função *Timing*, presente no próprio software. Os resultados encontram-se na Tabela 1.

Tabela 1 - Tempo de treinamento para cada método (em segundos).

Tentativa	Logistic Regression	Nearest Neighbors	Neural Network	Support Vector Machine	Naive Bayes
Tentativa #1	148,37	80,43	123,50	162,11	80,58
Tentativa #2	128,92	79,85	280,45	160,31	79,81
Tentativa #3	129,27	79,69	121,23	158,61	79,48
Tentativa #4	140,41	79,82	96,37	158,13	79,68
Tentativa #5	140,02	79,80	133,77	144,97	79,50
Tempo Médio	137,39	79,91	151,06	156,82	79,81

É possível configurar um parâmetro denominado *Performance Goal* à função *Classify*. Dessa forma, pode-se especificar uma meta de desempenho para o sistema, escolhendo-se o método que apresenta o melhor resultado em relação à meta estabelecida. As opções de metas de desempenho disponíveis são: qualidade, memória, tempo de treinamento e tempo de classificação. O tempo de treinamento difere do tempo de classificação na medida em que o primeiro é referente ao tempo que se leva para gerar

uma *ClassifierFunction* e o segundo é referente ao tempo que se leva para classificar um novo elemento. Os resultados encontram-se no Quadro 1.

Ao observar a Tabela 1, nota-se que os métodos *Nearest Neighbors* e *Naive Bayes* consumiram os menores tempos de treinamento, apresentando valores médios abaixo de 2 minutos, enquanto que os demais métodos apresentaram tempo superior. Contudo, confrontando-se com o Quadro 1, observa-se que o método *logistic regression* apresenta o melhor desempenho nas demais categorias, o que justifica e esclarece a sua escolha automática pelo software.

Quadro 1 - Métodos em função de metas de desempenho (*Performance Goal*).

	Qualidade	Memória	Tempo de Treinamento	Tempo de Classificação
Método	Logistic Regression	Logistic Regression	Nearest Neighbors	Logistic Regression

O terceiro teste consistiu na classificação das fotografias reservadas para o reconhecimento. A função classificadora calcula as probabilidades de pertencimento a cada classe, apresentando como resultado a classe associada à maior porcentagem. Dessa forma, o resultado é sempre uma classe, independentemente do valor percentual, o que pode se tornar um problema de implementação em um sistema de reconhecimento, uma vez que, na prática, o resultado de um reconhecimento é sempre positivo ou negativo. Porém, é possível especificar uma propriedade denominada *Indeterminate Threshold*, que estabelece um valor mínimo de porcentagem para que haja uma classificação. Se a maior porcentagem for menor do que o limiar, o resultado da classificação é indeterminado, o que soluciona o problema citado anteriormente. Os resultados do terceiro teste são apresentados na Tabela 2.

Tabela 2 – Reconhecimento das imagens armazenadas.

	Foto 1	Foto 2	Foto 3	Foto 4	Foto 5	Foto 6	Foto 7	Foto 8	Foto 9	Foto 10
%	94,87	85,26	30,27	40,58	70,12	44,58	78,40	76,24	93,70	98,91

Os resultados apresentados na Tabela 2 demonstram uma variação considerável nas probabilidades, desde valores muito baixos (30,27%) até valores elevados (98,91%). Além disso, a taxa de acerto foi de 90% (9 de 10 pessoas).

O teste final consistiu na aplicação prática do sistema completo, no qual é registrada uma fotografia do usuário, o reconhecimento é executado e o resultado é apresentado pelos dispositivos de saída. A fotografia do usuário não apresenta recorte automático de faces, uma vez que, como dito anteriormente, não é possível executar a função *FindFace* no *Raspberry Pi*. O valor limiar (*indeterminate threshold*) foi configurado em 0.7 (70%). Com isso, o teste final apresentou uma taxa de reconhecimento de 60%.

Diferente dos trabalhos relacionados que utilizaram um banco de imagens já processadas, nesta pesquisa foi criado um conjunto de fotos com pessoas realizando expressões faciais distintas e usando acessórios e variação da iluminação. Este banco heterogêneo, que representa a realidade de uso comumente encontrada em cenários que requerem controle de acesso, contribui para a redução da acurácia de sistemas de reconhecimento faciais.

CONCLUSÕES E TRABALHOS FUTUROS

Após a conclusão do trabalho proposto, observou-se que o tempo de treinamento foi expressivo, mesmo para um conjunto de dados de dimensão reduzida. Apesar disso, deve-se refletir sobre as situações em que o sistema será empregado, uma vez que o treinamento é realizado apenas uma vez a cada inicialização do sistema. Dessa forma, se o sistema permanecer ativo, o tempo gasto na fase de treinamento pode se tornar irrelevante.

Ademais, a dimensão do conjunto de dados é um fator que influencia diretamente o tempo de treinamento. Tendo em vista que, geralmente, num sistema de acesso seguro o número de usuários autorizados é restrito, o tempo de treinamento tende a diminuir.

Os resultados do teste prático refletiram algo que era esperado, uma vez que as fotografias do conjunto de dados apresentavam apenas os rostos recortados das pessoas, enquanto que as fotografias capturadas pelo sistema em execução não apresentavam esse recorte de faces.

A implementação do recorte automático de faces, do cadastramento de novos usuários e a integração do sistema a dispositivos avançados de acesso seguro constituem alternativas para trabalhos futuros, além de permitir o estudo de formas de aperfeiçoamento e aumento da eficiência do sistema apresentado.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAFRUIT. **Introducing the Raspberry Pi Model B+**. New York, Adafruit, 2015. Disponível em: <<https://learn.adafruit.com/introducing-the-raspberry-pi-model-b-plus-plus-differences-vs-model-b/overview>>. Acesso em: 02 jun. 2015.

CARVALHO, A. P. L. F. **Redes neurais artificiais**. São Paulo, USP, 2005. Disponível em <<http://www.icmc.usp.br/andre/research/neural>>. Acesso em 18 ago. 2015.

CONNOR, E. F. **Logistic Regression**. San Francisco: San Francisco University, [2015]. Disponível em: <<http://userwww.sfsu.edu/efc/classes/bio1710/logistic/logisticreg.htm>>. Acesso em: 18 ago.2015.

DELL. **Statistics: Methods and Applications**. Tulsa, Dell, 2013. Disponível em: <<http://documents.software.dell.com/Statistics/Textbook/Naive-Bayes-Classifer>> Acesso em: 18 ago. 2015.

FINK, Gernot A. **Markov models for pattern recognition: from theory to applications**. Springer Science & Business Media, 2014.

FOLHA DE SÃO PAULO. **Uberlândia (MG) adota reconhecimento facial para evitar fraudes no transporte coletivo**. São Paulo: Folha, 2014. Disponível em: <<http://www1.folha.uol.com.br/cotidiano/ribeiraopreto/2014/02/1407780-uberlandiamg-adota-reconhecimento-facial-para-evitar-fraudes-no-transporte-coletivo.shtml>>. Acesso em: 03 jun. 2014.

INFO ONLINE. **China desenvolve caixa eletrônico com reconhecimento facial**. São Paulo, Editora Abril, 2015. Disponível em: <<http://info.abril.com.br/noticias/seguranca/2015/06/china-desenvolve-caixa-eletronico-com-reconhecimento-facial.shtml>>. Acesso em: 01 jun. 2015.

G1 GLOBO. **Escola adota reconhecimento facial para controlar frequência de alunos**. Goiânia, Globo, 2015. Disponível em: <<http://g1.globo.com/goias/noticia/2015/05/escola-adota-reconhecimento-facial-para-controlar-frequencia-de-alunos.html>>. Acesso em: 20 mar. 2015.

MEHTA, S.; GUPTA, S.; BHUSHAN, B.; NAGPAL, C. K. **Face Recognition using Neuro-Fuzzy Inference System**. International Journal Of Signal Processing, Image Processing & Pattern Recognition, v. 7, n. 1, p.331–344, 2014.

NIKISINS, O.; FUKSIS, R.; KADIKIS, A.; GREITANS, M. **Face Recognition System on Raspberry Pi**. In: International Conference on Information Processing and Control Engineering (ICIPCE), 2015, Moscou – Rússia. Riga – Letônia: Institute of Electronics and Computer Science, 2015.

PANDYA, J. M.; RATHOD, D.; JADAV, J. J. **A survey of face recognition approach**. International Journal of Engineering Research and Applications (IJERA), v. 3, n. 1, p. 632–635, 2014.

PEDREGOSA, Fabian et al. **Scikit-learn: Machine learning in Python**. Journal of Machine Learning Research, v. 12, n. Oct, p. 2825-2830, 2011.

RIBEIRO, Í.; CHIACHIA, G.; MARANA, A. N. **Reconhecimento de Faces Utilizando Análise de Componentes Principais e a Transformada Censur**. In: Workshop de Visão Computacional, 6., 2010, Presidente Prudente. *Anais eletrônicos*. Presidente Prudente: FCT/UNESP, 2010.

SHEN, Y.; HU, W.; YANG, M. **Face recognition on smartphones via optimised sparse representation classification**. IPSN '14 Proceedings of the 13th international symposium on Information processing in sensor networks. Anais... p.237–248, 2014. IEEE Press.

WOLFRAM. **Documentation Center**. Champaign, Wolfram [2015]. Disponível em: <<http://reference.wolfram.com/language/?source=nav>>. Acesso em: 02 jun. 2015.