DESTAQUES ACADÊMICOS
UNIVATES | CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Revista Destaques Acadêmicos, Lajeado, v. 12, n. 4, 2020. ISSN 2176-3070
DOI: http://dx.doi.org/10.22410/issn.2176-3070.v12i4a2020.2740
http://www.univates.br/revistas

CONTROLE DE DISPOSITIVOS EM CASAS INTELIGENTES POR MEIO DO RECONHECIMENTO DE GESTOS E BIOMETRIA FACIAL COM USO DE DEEP LEARNING

Willian Valmorbida[1], Klaus Fernando Etgeton[2]

Resumo: Estima-se que bilhões de dispositivos IoT estarão conectados nos próximos anos, criando inúmeras possibilidades para automação em casas inteligentes que poderá tornar a vida das pessoas mais prática, eficiente, confortável. Além das funcionalidades, as pessoas também esperam que as tecnologias se integrem e façam parte do ambiente, proporcionando melhores experiências de interação de maneira mais fácil e natural. Neste trabalho, foi desenvolvido um sistema de visão computacional em Python, com uso de modelos em Deep Learning, para controlar dispositivos IoT conectados em casas inteligentes por meio do reconhecimento de gestos estáticos da mão, realizando uma interação mais segura por meio da autenticação biométrica facial dos usuários. Para o reconhecimento em tempo real, o sistema apresenta um frame rate de 3 FPS (Frames Per Second). O reconhecimento facial dos usuários apresentou excelentes resultados, com apenas 1.10% de falso positivo, enquanto o reconhecimento de gestos apresentou 22.67%. O sistema proposto atende os requisitos necessários com desempenho relativamente bom, porém melhorias quanto ao reconhecimento de gestos ainda precisam ser realizadas para reduzir sua taxa de falso positivo.

Palavras-chave: Visão computacional. Casas inteligentes. Biometria facial. Reconhecimento de gestos. Deep Learning.

1. Introdução

Pode-se observar que nos últimos anos houve um investimento considerável em pesquisas e desenvolvimento de produtos que utilizam da visão computacional para a identificação, reconhecimento e classificação de informações em imagens. Há inúmeros projetos de pesquisa e ferramentas que propõem solucionar e criar modelos que consigam classificar e identificar desde pessoas, animais, placas de trânsito e até mesmo sentimentos.

1 Mestre em Computação Aplicada – Unisinos-RS e professor do Centro de Ciências Exatas e Tecnológicas da Universidade do Vale do Taquari – Univates.

2 Bacharel em Engenharia de Software – Univates-RS.

A visão computacional possui inúmeras aplicações, tanto na indústria, ao realizar a inspeção de defeitos em placas eletrônicas na linha de produção, quanto no meio domiciliar, presente nas interfaces dos videogames, em câmeras e celulares ao remover os olhos vermelhos das pessoas presentes nas fotos (DAWSON-HOWE, 2014).

Tecnologias provenientes da visão computacional estão proporcionando diferentes meios para que possamos interagir e facilitar tarefas cotidianas, proporcionando redução de tempo e praticidade. A criação de ambientes inteligentes e com maneiras práticas de interação se fazem possíveis graças ao aprimoramento de tecnologias portáteis e de baixo custo. Dispositivos dos mais variados tipos agora possuem uma camada de comunicação que possibilita desde ações mais simples como ligar e desligar a até mais complexas como controlar a central de uma casa inteligente.

Até o ano 2021, estima-se que o número de dispositivos de Internet Of Things (IoT) conectados alcançará o patamar de 25 bilhões, estes dispositivos tornarão nossa vida mais prática, eficiente, econômica e confortável (GARTNER, 2018). Com o uso de assistentes pessoais treinadas com Machine Learning é possível que tarefas do cotidiano como reserva em restaurantes e controle das luzes em casa sejam automatizados.

Este trabalho apresenta o desenvolvimento de um sistema para controle de dispositivos IoT conectados, em ambientes de casas inteligentes, por meio do reconhecimento de gestos estáticos da mão, garantindo a segurança por meio da autenticação da biometria facial.

2. Referencial Teórico

Para o desenvolvimento da solução proposta, torna-se necessário o estudo de visão computacional para processamento de imagens, bem como modelos de identificação, extração da biometria facial e *machine learning*. Estes assuntos serão abordados nas seções a seguir, visando exemplificar o contexto e os problemas que este trabalho se propõe a resolver.

2.1 Visão computacional

Conforme Dawson-Howe (2014, p. 19), visão computacional é a análise automática de imagens e vídeos a fim de obter uma compreensão do mundo. É baseada na capacidade de visão humana, sendo inicialmente abordada como tema na década de 60 e 70, pensada como um problema simples de ser resolvido, devido ao nosso próprio sistema visual que faz o processamento parecer algo intuitivo para a nossa mente. No entanto, o sistema visual humano é muito complexo e estima-se que nosso cérebro utiliza de 25% até mais de 50% de nossa capacidade para processar as imagens, realizando inúmeras interpretações ao visualizar uma imagem, como sentimentos, o vento sobre as folhas, interação entre pessoas.

2.1.1 Equalização do histograma de cores de uma imagem

Há situações durante a visualização de imagens onde que há regiões muito claras e outras regiões da mesma muito escuras, há uma grande variedade de tonalidades que dificultam a compreensão de uma imagem, entretanto se for realizado a distribuição entre as escalas em cinza podemos compreender de maneira melhor o que está presente na imagem. Uma técnica que realiza esta operação de distribuição das escalas em cinza em uma imagem é a equalização do histograma, que permite obter como resultado uma imagem com maior contraste e níveis de luminosidade equilibrados (DAWSON-HOWE, 2014).

2.1.2. Segmentação

A segmentação de imagem é o processo de quebrar ou segmentar uma imagem em múltiplos grupos de *pixels* que geralmente possuem características em comum, tendo como principal objetivo representar a imagem de um modo que faça mais sentido e consequentemente facilitando sua análise. O processo de segmentação também pode ocorrer com base na similaridade de cor ou forma do conjunto de *pixels* (MATHWORKS, 2019).

O método mais simples de segmentação é o Thresholding que realiza a separação das regiões com base na variação de intensidade entre os *pixels* do objeto de interesse e os *pixels* do fundo da imagem. Para diferenciar os *pixels* do objeto de interesse é realizado a comparação da intensidade de cada *pixel* a um valor limite (*threshold*) (OPENCV, 2019).

2.1.3 Detecção de objetos com classificadores

O processo de detecção de objetos ou formas é comumente utilizado em inúmeras tarefas da computação para encontrar faces, pessoas, placas de trânsito. O algoritmo Haar proposto por Paul Viola e Michel Jones se demonstra como uma maneira efetiva para resolver o problema de detecção de objetos. O algoritmo é baseado em Machine Learning e para a criação de um modelo classificador é necessário o treinamento com uma grande quantidade de imagens positivas e negativas. Para a aprendizagem deste modelo são necessárias imagens positivas, ou seja, que representam o objeto de interesse de detecção e imagens negativas onde não há o objeto de interesse, desta maneira o modelo aprende a considerar quais são as características do objeto em questão (OPENCV, 2019).

2.2 Biometria

Segundo Nakashiro (2011), para se diferenciar uma pessoa de outra é necessário que exista um meio de definir sua identidade, ou seja, um conjunto de características únicas que se diferem de outra pessoa, como traços físicos, traços faciais, a íris, a retina, voz e grafia.

Os sistemas de identificação que utilizam da característica biométrica como fator de autenticação, dificilmente conseguem 100% de certeza, pois algumas de nossas características biométricas podem variar e sofrer pequenas alterações ao longo de nossa vida, dificultando a comparação do padrão biométrico extraído. Como agravantes ao realizar a identificação de uma pessoa, Moraes (2010) menciona, na identificação facial a presença ou ausência de barba, óculos, acessórios. Frente à variação, é necessário definir um grau de certeza para determinar se uma pessoa será ou não autenticada.

Segundo Moraes (2010), as soluções mais populares que resolvem o problema de reconhecimento facial, baseiam-se na localização e análise de atributos faciais como nariz, olhos, ou da análise geral da face. O fluxo básico de sistemas de biometria facial possui quatro etapas básicas, são elas:

1. Detecção da existência de face em imagem ou em vídeo;
1. Localização da face na imagem;
2. Extração das características faciais (posição nariz, boca, olhos) e comparação com base de conhecimento;
3. Retorno do padrão armazenado mais próximo do padrão de assinatura biométrico da imagem de entrada.

As tecnologias de reconhecimento facial ainda possuem o desafio de resolver problemas como iluminação e posição da face em sentido de rotação do rosto. Por situações como estas, sistemas de biometria facial ainda possuem desempenho relativamente baixo se comparados com sistemas de impressão digital e íris (TOLBA; EL-BAZ; EL-HARBY, 2008).

2.3 Reconhecimento facial

Dentre as diferentes propostas para reconhecimento de faces em imagens, o algoritmo baseado em Histogram of Oriented Gradients (HOG), destaca-se por ser capaz de extrair características de imagens por meio da diferença da orientação dos gradientes existentes (DALAL; TRIGGS, 2005). A detecção facial com a utilização de algoritmo em HOG inicia com a conversão da imagem para preto e branco, pois não é necessário o maior detalhamento de cores para este algoritmo. Para cada *pixel* da imagem é realizado uma comparação com os *pixels* que o cercam, identificando quanto o *pixel* atual é mais escuro aos *pixels* a sua volta. Neste processo são definidas linhas no sentido em que a imagem se torna mais escura (GEITGEY, 2016).

Este processo de criação de linhas resulta em um gradiente da imagem que mostra todo o fluxo do branco para o preto, visando resolver a complexidade de detectar faces. Para resolver o grande detalhamento deste gradiente a imagem é novamente quebrada em blocos de 16x16 *pixels*, realizando um processo similar ao anterior, que define o principal sentido dos gradientes dentro do bloco (GEITGEY, 2016).

Com a imagem convertida em HOG é realizado a comparação com um padrão HOG criado através de inúmeras faces, assim se houver uma face na imagem com um padrão similar ao padrão HOG facial, a face será definida como detectada (GEITGEY, 2016).

2.4 Machine Learning

Machine Learning se faz interessante em resolver problemas em que dependendo da complexidade do padrão a ser detectado, um humano não conseguiria definir uma regra ou um conjunto explícito de regras que poderiam ser utilizadas na classificação dos dados, porém com a utilização de diferentes algoritmos, a máquina é capaz de aprender a relação dos dados e assim, inferir respostas (SHALEV-SCHWARTZ; BEM-DAVID, 2014).

2.4.1 Deep Learning

Deep Learning é uma subseção de Machine Learning, focada em algoritmos inspirados na estrutura e funcionamento do cérebro, denominados de redes neurais artificiais. As redes neurais artificiais são construídas de modo similar a um cérebro humano, onde há nós de neurônios que se conectam como uma rede, de maneira hierárquica, o processamento de uma camada de neurônios serve como fonte de entrada para a camada posterior, a cada avanço as informações se tornam mais especializadas. Este modelo de estrutura possibilita que a aprendizagem ocorre de maneira não linear, diferente da maneira tradicional do funcionamento de algoritmos baseados em Machine Learning (BROWNLEE, 2019).

Para o treinamento em grandes conjuntos de dados o modelo tradicional de Machine Learning não é eficiente, pois há o problema de como escalar a aprendizagem, cada vez mais informações surgem e o desempenho não aumenta conforme mais informações são inseridas, em contrapartida Deep Learning possibilita a escalabilidade do processamento (NG, 2015).

Algoritmos baseados em Deep Learning podem ser aplicados para inúmeras áreas, como visão computacional, reconhecimento de imagens, sistemas de reconhecimento de voz, reconhecimento de padrões (AJIT, 2015).

O processamento de imagens é altamente sensível e dependente da localização das características que existem em uma região, para que se possa inferir e classificar informações em uma imagem (BROWNLEE, 2019). Para abordar este problema pode ser utilizado Convolutional Neural Network (CNN), um algoritmo capaz de representar as características que existem em uma imagem, atribuindo importância à vários elementos que existem nela e também sendo capaz de diferenciar uns dos outros. O pré-processamento necessário em uma *convolutional network* é muito menor se comparado com outros métodos de classificação, este algoritmo é capaz de aprender a classificar e filtrar as características se corretamente treinado (SAHA, 2018).

3. Metodologia

O presente trabalho apresenta o desenvolvimento de um sistema que permite a interação com dispositivos IoT por meio da identificação da biometria facial e reconhecimento dos gestos estáticos. Para o desenvolvimento do sistema proposto diferentes estratégias foram pensadas para criar uma solução viável, diferentes tecnologias foram exploradas e testadas ao decorrer do processo, desta forma, o presente trabalho se caracteriza como uma pesquisa exploratória, que conforme Gil (2006), busca criar uma maior base de conhecimento sobre o problema, incentivando a criação de hipóteses e soluções.

Gil (2006) cita que pesquisas exploratórias são bastante flexíveis, possibilitando um escopo dinâmico, conforme a necessidade. A pesquisa pode ser composta por uma revisão bibliográfica, entrevistas com pessoas que tiveram experiências relacionadas e análise de exemplos que clarifiquem a compreensão do problema.

A partir do estudo bibliográfico e a análise do estado atual dos algoritmos de visão computacional foi especificado o projeto e seus requisitos. Com base na definição do projeto foi realizado desenvolvimento de um protótipo no qual foram realizados testes a fim de validar a viabilidade desta solução, foram analisados os resultados com base nos seguintes critérios: identificação do gesto, assertividade da biometria facial e o tempo de resposta final.

4. Desenvolvimento

Neste capítulo são apresentados detalhes do desenvolvimento do protótipo, assim como detalhes sobre as bibliotecas e tecnologias utilizadas para implementação do reconhecimento de gestos, biometria facial e controle de *protoboards*.

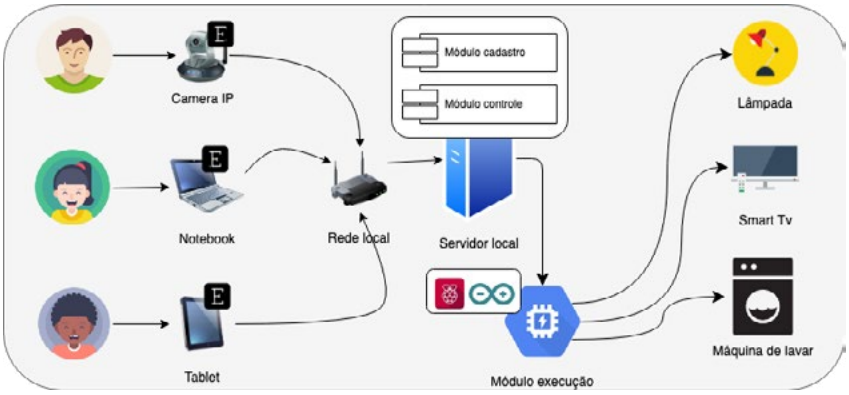
4.1 Visão geral

Para validar a solução proposta, foi desenvolvido um protótipo, onde que todos os módulos do protótipo foram disponibilizados em um notebook, com 4GB de RAM e com processador i5 com dois núcleos independentes. Para a captura das imagens processadas, foi utilizado a câmera integrada. Foram empregadas no desenvolvimento do protótipo as tecnologias: Python, OpenCV, NodeJS, Arduino (para o módulo de execução) e JoHnny-Five.

A Figura 1 apresenta o fluxo de interação dos módulos e atores da arquitetura. Diferentes pessoas dentro de um contexto compartilhado, como em uma casa inteligente, se conectam e interagem com a aplicação através de fontes de entrada de imagem e vídeo, assim representados pela letra E (Entrada). Estes dispositivos de entrada quando conectados são consumidores do serviço, assim reconhecidos como clientes do sistema. Cada cliente é conectado através da rede com o módulo Controle que realiza o processamento

de visão computacional, assim como a identificação facial e o reconhecimento de gestos. Ao identificar uma pessoa autorizada e o gesto realizado pela mesma, é disparado uma comunicação com o módulo Execução que se encarrega de executar a ação correspondente com os diferentes dispositivos conectados, como lâmpadas, máquinas de lavar e, *smart* TV.

Figura 1 - Fluxo de interação e comunicação do projeto.



Fonte: Elaborado pelo autor (2019).

Ao iniciar, o módulo de Controle carrega o algoritmo para reconhecimento facial e de gestos, juntamente com o modelo já treinado para reconhecimento dos usuários e dos gestos. O algoritmo então realiza a captura de cada *frame* da câmera, redimensiona para o tamanho de 600x600 pixels. Este tamanho foi determinado por considerar-se suficiente para a aplicação pretendida e porque resoluções superiores impactam no desempenho da aplicação.

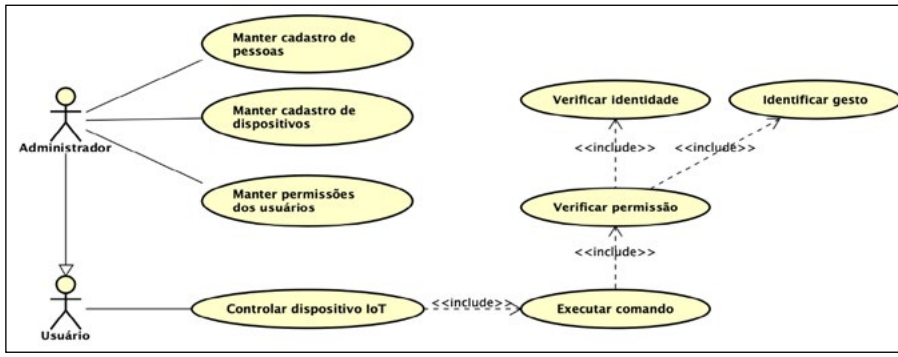
O módulo de Controle realiza a captura e processamento dos *frames*, que consiste na identificação facial e o reconhecimento do gesto. Após a identificação correta de um usuário cadastrado é realizado a identificação do gesto e a comunicação com o módulo de Execução por meio de chamadas REST.

O módulo de Execução disponibiliza um *webservice* REST em NodeJS, como um *gateway* de comunicação com diferentes dispositivos, sendo realizado a comunicação com a *protoboard* Arduino Uno para o acionamento das saídas digitais conforme configuração enviada pelo módulo de Controle.

4.2 Casos de uso

Para o administrador do sistema dentro de um contexto controlado, como em uma residência, há a necessidade de cadastrar as pessoas que poderão controlar o ambiente e também especificar quais dispositivos e ações cada pessoa terá acesso para comandar o ambiente, conforme apresentado na Figura 2.

Figura 2 - Casos de uso do sistema.



Fonte: Elaborado pelo autor (2019)

Depois de realizado o cadastro das pessoas, é executado o treinamento com modelo baseado em Deep Learning para extração da biometria facial das múltiplas fotos de cada usuário. Assim quando algum usuário realizar gestos para controlar algum dispositivo, é verificado a permissão, que ocorre através da biometria facial e consequentemente são identificadas as ações permitidas, como apresentado no caso de uso da Figura 2. Assim que as devidas identificações são confirmadas, é executado a comunicação com o dispositivo IoT, através do módulo Execução.

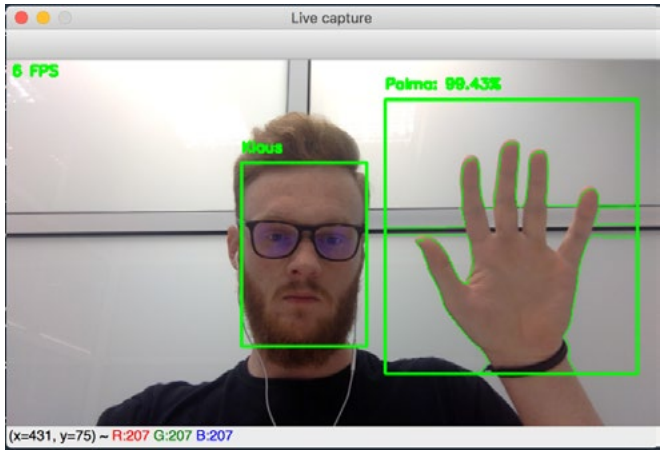
4.3 Interface do protótipo

A interface visual do protótipo foi desenvolvida para o propósito de verificação dos resultados e desempenho geral da solução. Na interface do protótipo é possível visualizar a detecção das principais áreas de interesse (Region of Interest - ROI) para o processamento do algoritmo de visão computacional, como a área onde que a face e a mão se encontram.

Para a ROI facial detectada na interface também é exibido o nome do usuário se corretamente identificado como válido, em caso de não reconhecimento a face é definida como "Desconhecida".

Além da identificação da face, também é possível visualizar a ROI com a detecção do gesto quando o mesmo for identificado como conhecido. A interface do protótipo, assim como os reconhecimentos realizados e a medição do desempenho de 5 FPS pode ser visualizada na Figura 3.

Figura 3 – Interface com identificação facial, gesto e velocidade do processamento



Fonte: Elaborado pelo autor (2019).

Para realizar o cadastro dos usuários no protótipo, foi desenvolvido um algoritmo, juntamente com a interface visual do protótipo, onde que é possível iniciar a inserção de um novo usuário mediante o pressionamento da tecla "N", assim o algoritmo irá esperar até que o nome para identificação do novo usuário seja informado, ao confirmar o nome, o sistema iniciará a captura da face da pessoa, sendo que para cada cadastro são capturadas múltiplas imagens da mesma pessoa. São armazenados em uma pasta específica somente a ROI, com a face centralizada de cada imagem do novo usuário. O treinamento das faces cadastradas ocorre mediante a execução manual de um *script* em Python que inicia o treinamento, realizando a extração das características faciais como fonte de entrada para o algoritmo de treinamento, como o Fisherfaces do OpenCV ou o modelo Deep Learning com o *framework* Dlib. Para que os novos usuários possam ser identificados em tempo real é necessário que o sistema seja inicializado novamente após a realização do treinamento.

4.4 Detecção e Reconhecimento Facial

A detecção facial foi implementada fazendo uso do algoritmo baseado em Histogram of Oriented Gradients (HOG) do *framework* Dlib. O processo para a detecção facial inicia com a conversão da imagem para preto e branco (não é necessário o maior detalhamento de cores para este algoritmo) e na imagem convertida é aplicado o algoritmo de detecção, sendo obtidos como resultado as ROI de todas as pessoas presentes no *frame* e para cada ROI é realizado o reconhecimento facial que é detalhado na seção seguinte.

Durante o desenvolvimento da funcionalidade de identificação facial do protótipo foram realizados testes com dois algoritmos distintos, com o objetivo de obter o melhor resultado e a menor taxa de falso negativo e falso positivo.

O primeiro algoritmo utilizado foi o Fisherfaces disponibilizado pelo OpenCV, porém, o melhor resultado foi obtido utilizando o modelo de Deep Learning para extração das características faciais através do *framework* Dlib.

Dois usuários foram utilizados para a validação do protótipo, sendo que quantidades diferentes de fotos foram utilizadas para o treinamento de cada usuário, para o primeiro foram utilizadas 129 fotos e 58 para o segundo.

A utilização deste algoritmo baseado em Deep Learning ocorreu de maneira mais simples, se comparado com a identificação realizada com o OpenCV, pois os resultados até então já eram superiores e adequados para a homologação do protótipo, sendo assim a etapa de normalização não foi realizada. O preparo de cada imagem e treinamento ocorreu conforme as seguintes etapas:

1. Conversão da imagem para o esquema de cores preto e branco;
2. Detecção da ROI - posição da face;
3. Face detectada em esquema de cores RGB é inserida e o processo de aprendizagem é realizado com o reconhecedor baseado em Deep Learning;
4. Os dados extraídos do reconhecimento facial são salvos separadamente para a posterior utilização do modelo ensinado.

O algoritmo disponibilizado pelo *framework* Dlib é baseado em um modelo de Deep Learning que foi treinado em uma base de imagens em torno de 3 milhões de faces para ser capaz de extrair e definir 128 medidas denominadas de *embeding* que quantificam as características faciais de uma pessoa. O treinamento deste algoritmo ocorre por um processo conhecido como Triplet Training Step, onde que por meio de três imagens únicas, sendo duas da mesma pessoa e outra de uma pessoa diferente, é criado o vetor dimensional de 128 pontos. O peso entre a distância das características faciais das duas fotos da mesma pessoa é aproximado e terceira face diferente serve para distanciar os valores do que seria o conjunto de dados de uma pessoa da outra (GEITGEY, 2016).

Este algoritmo, diferente do utilizado no Fisherfaces do OpenCV, utiliza o esquema de cores em formato RGB, não sendo necessário a conversão para preto e branco, sequer o ajuste de contraste. A ROI com a face de cada usuário não passou pelo processo de normalização que foi realizado no treinamento com o algoritmo Fisherfaces. O *embeding* facial da face é extraído e comparado com o modelo previamente treinado no módulo Cadastro, o resultado do reconhecimento facial deste algoritmo, diferente dos reconhecedores do OpenCV, retorna uma lista de candidatos, sendo necessário filtrar pelo candidato que recebeu mais votos para se obter a identidade da pessoa. Este algoritmo também se diferencia por não necessitar um limite *threshold*, a recomendação para o reconhecimento leva em consideração que se a distância euclidiana do vetor da face da pessoa for menor que 0.6 em comparação com outro vetor

então os vetores são da mesma pessoa. No protótipo foi utilizado como base 0.5 para a distância euclidiana ao realizar a comparação e identificação dos usuários, este parâmetro é definido ao solicitar o reconhecimento da face, o valor padrão configurado é de 0.6 para a distância euclidiana máxima, assim utilizando valores inferiores a rigorosidade do reconhecimento é aumentada.

4.5 Reconhecimento de gestos

Para o reconhecimento de gestos foi feito o uso de um projeto público, chamado "Hand Gesture Recognition Using Background Ellimination and Convolution Neural Network" (HGRUBECNN), que é baseado em Convolution Neural Network (CNN), uma rede neural que realiza o reconhecimento de gestos estáticos das mãos (SAHA, 2019). O projeto em questão possui uma rede neural configurada, não observando-se necessidade de modificação de sua configuração padrão. No entanto, devido a diferença entre as versões dos pacotes em Python do protótipo desenvolvido e do projeto HGRUBECNN, foi necessário que o treinamento fosse executado novamente.

O projeto HGRUBECNN inclui imagens para treinamento e modelo para realizar o reconhecimento de três gestos distintos, sendo eles: punho, palma da mão e *swing*, respectivamente. Para que seja possível reconhecer mais gestos é necessário que sejam preparadas imagens dos gestos desejados e executado novamente o treinamento, assim sendo possível utilizar mais gestos para a interação.

Uma etapa de extrema importância para o reconhecimento de gestos é a segmentação da ROI, visando que o fundo seja abstraído e apenas a região da mão fique visível, assim sendo possível remover detalhes da imagem que poderiam atrapalhar o treinamento e o reconhecimento (MATHWORKS, 2019). Porém, o projeto HGRUBECNN utiliza segmentação baseada em *binary thresholding* que sofre grande influência das condições de luminosidade presentes no ambiente. Para casos de diferentes luminosidades seria mais adequado a utilização de *adaptive thresholding*. O algoritmo para realização da segmentação com *adaptive thresholding* está disponível pela API do OpenCV, porém não foi possível adaptar esta etapa do reconhecimento implementado no projeto HGRUBECNN pois as imagens para treinamento foram convertidas utilizando *binary thresholding* e as imagens originais não estão disponíveis para que novos testes com um *thresholding* diferente seja facilmente utilizado (OPENCV, 2019).

O projeto HGRUBECNN não realiza a detecção da mão, mas a captura de uma região fixa nos *frames*. Para avançar neste sentido, foi adicionado ao protótipo, o uso de dois classificadores Haar, publicamente disponíveis pelo OpenCV. Eles servem para a detecção de quando a mão está aberta e fechada, assim o processo de detecção da posição da mão no pior cenário pode necessitar que os dois classificadores sejam executados, pois se não é possível detectar

a mão com o primeiro classificador, é realizado novamente o processo com o segundo e se não forem encontrados resultados assume-se de que não existe uma mão na região do *frame* em processamento.

Como etapa final de normalização da ROI, é realizado um aumento de cerca de 20 *pixels* em torno da região detectada, pois o tamanho da mão ao realizar o gesto pode ser maior do que a ROI detectada pelos classificadores em cascata, assim evita-se de que a imagem fique cortada e não seja possível de identificar o gesto realizado. Por final a ROI é redimensionada para 89x100 pixels, formato esperado para a realização da detecção do gesto pelo modelo de reconhecimento baseado em CNN. Como resultado do reconhecimento há dois valores, o *id* da classe referente ao gesto e em segundo lugar o valor que define a taxa de assertividade denominada de *confidence*, assim assume-se como verdadeiro os resultados acima de 0.99, sendo a assertividade máxima possível definida pelo valor 1.

4.6 Acionamento de dispositivos

O acionamento de dispositivos à cargo do módulo de Execução foi implementado através de um serviço REST desenvolvido em NodeJS, gerenciando o recebimento das chamadas HTTP e como resultado final é realizado o acionamento ou desligando de algum dispositivo conectado às saídas digitais ou analógicas de um Arduino Uno.

Para a validação do protótipo foi feito o uso da placa *protoboard* Arduino Uno conectada por meio da USB, porém outras placas poderiam ser utilizadas. O interessante neste módulo é a utilização do *framework* Johnny-Five que fornece uma camada de abstração para desenvolvimento de robótica e também plataformas IoT, assim de maneira simples, é possível que diferentes *protoboards* possam ser utilizadas, seguindo uma API única de comunicação.

Para a validação deste protótipo e gerenciamento das ações provenientes dos gestos reconhecidos, foram utilizados apenas dois gestos. Para o acionamento foi utilizado o gesto referente ao estado de mão aberta e para o desligamento o gesto onde que a mão se encontra fechada, não sendo necessário a utilização do terceiro gesto *swing*. Como teste deste sistema foi definido que as interações dos gestos controlariam um LED, sendo evidente o acionamento e desligamento por meio de seu sinal luminoso.

5. Avaliação do protótipo e resultados

Neste capítulo serão apresentados os resultados obtidos a partir do protótipo desenvolvido. Para validação dos algoritmos de reconhecimento foram coletadas e analisadas novas imagens, diferentes das utilizadas para o treinamento dos algoritmos, a fim de medir suas assertividades. Por fim, foi realizada a medição do tempo médio para processamento individual dos algoritmos e o tempo total médio para processamento de cada *frame*. Com

exceção da Amostra 3 do reconhecimento facial, todos os testes a seguir foram realizados em ambiente controlado de luminosidade e com pelo menos dois usuários.

Para a captura correta das imagens é necessário que a câmera seja posicionada corretamente no local de interesse, pois a interação em situações de demasiado longe, acima de 2.5 metros da câmera, ou demasiado perto, menos de 1 metro da câmera, não é possível obter bons resultados. Para a situação de maior distância, o reconhecimento de gestos perde seu percentual de *confidence*, não atingindo o mínimo esperado para aceitar o gesto. E para a situação muito próxima à câmera não se recomenda, pois os classificadores para detecção da mão não reconhecem corretamente a mesma, uma vez que muito próxima, a mão tem um tamanho maior do que o esperado para detecção dos classificadores. Alteração para detecção da mão é possível, porém impactaria também desempenho final da solução.

5.1 Reconhecimento facial

Nesta seção serão analisados os resultados dos testes realizados com os algoritmos de biometria facial utilizados no protótipo, foram coletadas e testadas 3 amostras de imagens (Tabela 1), cada imagem das amostras foi submetida aos dois modelos de reconhecimento facial treinados durante o desenvolvimento do protótipo.

Tabela 1 – Resultado das Amostras – reconhecimento do primeiro usuário

Amostra 1	Deep Learning com Dlib	Fisherfaces com OpenCV
Verdadeiro positivo	100	75
Falso negativo	0	25
Amostra 2	Deep Learning com Dlib	Fisherfaces com OpenCV
Verdadeiro positivo	100	16
Falso negativo	0	84
Amostra 3	Deep Learning com Dlib	Fisherfaces com OpenCV
Verdadeiro negativo	269	225
Falso positivo	3	47

Fonte: Elaborado pelo autor (2019).

Para a amostra 1 foram coletadas 100 imagens faciais de um usuário, sendo estas imagens diferentes das 58 imagens utilizadas para seu treinamento. O teste para esta amostra aponta melhores resultados para o modelo baseado em Deep Learning, onde 100% da amostragem foi identificada corretamente, para o algoritmo Fisherfaces houve uma taxa de 25% de falso negativo, causando a não identificação do usuário.

Para a amostra 2 também foram coletadas 100 imagens faciais do segundo usuário, sendo estas imagens diferentes das 129 imagens utilizadas para seu

treinamento. Os testes realizados sobre esta amostra apontam novamente melhores resultados para o modelo baseado em Deep Learning, onde 100% da amostragem foi identificada corretamente, já para o algoritmo Fisherfaces houve um alto valor para falso negativo. É interessante ressaltar que apesar do maior número de imagens utilizadas para treinamento se comparado com o usuário da Amostra 1 os resultados não foram superiores.

Para a amostra 3 foram utilizadas 272 imagens faciais de pessoas desconhecidas e com diferentes condições de luminosidade, tendo como objetivo a identificação dos melhores resultados para verdadeiro negativo e falso positivo, assim sendo possível verificar qual algoritmo obteve melhores resultados na identificação de pessoas que não pertencem ao conjunto de usuários cadastrados e também se o algoritmo reconhece impostores como usuários verdadeiros do sistema. Com base nos resultados é possível identificar que o algoritmo baseado em Deep Learning obteve excelentes resultados na identificação correta de usuários impostores do sistema, com apenas uma pequena taxa de 1.10% para falso positivo. Enquanto o algoritmo Fisherfaces obteve um alto valor para o reconhecimento de impostores como usuários válidos do sistema, com uma taxa de 17.28%.

5.2 Reconhecimento de gestos

Nesta seção serão analisados os resultados dos testes com o algoritmo de reconhecimento de gestos utilizado no protótipo, foram coletadas e testadas 3 amostras de imagens diferentes das utilizadas para o treinamento, conforme resultados apresentados na Tabela 2.

Tabela 2 – Resultados das Amostras

Amostra 1 - Palma da mão	Resultados
Verdadeiro positivo	100
Falso negativo	0
Amostra 2 - Punho da mão	**Resultados**
Verdadeiro positivo	76
Falso negativo	24
Amostra 3 - Gestos desconhecidos	**Resultados**
Verdadeiro negativo	232
Falso positivo	68

Fonte: Elaborado pelo autor (2019).

Para a amostra 1 foram coletadas 100 imagens onde a palma da mão era o gesto realizado. Conforme os testes, foi identificado alta taxa de reconhecimento para a palma da mão, porém a eficácia deste resultado será discutida melhor na Amostra 3.

Para a amostra 2 foram coletadas 100 imagens onde o punho da mão era o gesto realizado. Diferente do reconhecimento da palma da mão, o modelo não foi capaz de reconhecer todas as imagens desta amostra como verdadeiros, apresentando uma taxa de 24% para falso negativo.

Para a amostra 3 foram coletadas 300 imagens onde o gesto da mão é desconhecido para o modelo, assim sendo possível verificar se o modelo reconhece corretamente gestos inválidos como tais e não pertencentes a nenhum gesto conhecido do modelo.

Conforme os resultados obtidos, fica evidente o alto número para falso positivo, ou seja, o modelo ainda não é capaz de distinguir de maneira adequada gestos desconhecidos dos gestos treinados e consequentemente há uma incorreta classificação de um gesto inválido como algum válido do modelo, apresentando uma taxa de 22.67% para falso positivo. As imagens utilizadas para o teste possuem em sua maioria pelo menos algum dos dedos em evidência, assim o modelo acabou reconhecendo gestos com mais de um dedo em evidência como a palma da mão. Assim o resultado da Amostra 1 se torna mais evidente, o modelo acaba sendo tendencioso a classificar algum gesto com mais dedos como o sendo gesto da palma da mão onde todos os 5 dedos estão visíveis.

5.3 Desempenho geral do protótipo

O desempenho geral do protótipo foi calculado com base em uma média de 200 execuções, os valores podem ser visualizados na Tabela 3.

Tabela 3 - Tempo para processamento de cada algoritmo do protótipo

Etapa	Tempo (milissegundos)
Reconhecimento facial	~162
Reconhecimento de gestos	~58
Execução de dispositivo	~11
Tempo total	~291

Fonte: Elaborado pelo autor (2019)

Quanto ao tempo de processamento, o protótipo apresentou bons resultados, sendo possível completar o processamento de pelo menos 3 *frames* por segundo. Tanto a identificação facial quanto o reconhecimento de gestos ocorrem a cada *frame* processado, isto é possível pelo fato de ser realizado o reconhecimento de gestos estáticos das mãos e não gestos em movimento que demandariam o processamento de múltiplos *frames* em sequência. O consumo dos recursos de *hardware* para executar o protótipo foi de 100% de um processador e 10% do segundo, com uma utilização de 400 MB de memória RAM.

6. Conclusões

Com o objetivo de desenvolver um sistema de interação com dispositivos em casas inteligentes, o presente trabalho apresentou o desenvolvimento de um protótipo que possibilita a interação com dispositivos conectados por meio do reconhecimento de gestos das mãos, garantindo segurança durante a interação por meio da identificação biométrica facial dos usuários.

Através do desenvolvimento do protótipo foi possível integrar diferentes algoritmos de diferentes tecnologias para o desenvolvimento de uma solução para interação com o ambiente, como em casas inteligentes. Através dos testes realizados foi possível constatar excelentes resultados do modelo baseado em Deep Learning para reconhecimento facial se comparado ao algoritmo de reconhecimento facial Fisherfaces. O protótipo desenvolvido apresentou bons resultados de desempenho para o processamento em tempo real de cada *frame*, sendo possível realizar o processamento completo de pelo menos 3 *frames* por segundo.

As principais dificuldades durante o desenvolvimento do protótipo estiveram relacionadas às estratégias para abordar e solucionar os problemas de visão computacional em geral, como segmentação, detecção correta das ROI e qualidade dos resultados obtidos. Além disto, foi necessário realizar diferentes testes com diferentes resoluções na busca de um bom desempenho sem perda na qualidade do reconhecimento facial, embora no final, os piores resultados estiveram relacionados ao reconhecimento de gestos, onde foi necessário a utilização de diferentes classificadores para detectar as mãos e a busca por modelos capazes de reconhecer corretamente os gestos realizados.

Embora a abordagem satisfaça os objetivos propostos pelo trabalho, existem melhorias que poderiam ser desenvolvidas e algumas estratégias repensadas para o desenvolvimento de uma solução com maior desempenho, usabilidade no cadastro de usuários e precisão no reconhecimento de gestos.

Pelo fato de o trabalho proposto demandar o estudo e conhecimento de diferentes tecnologias, até o momento alcançou-se o desenvolvimento de um protótipo funcional, que necessita de trabalhos futuros para sua finalização.

Por questões de segurança é necessário de que quando existam mais de uma pessoa em uma mesma ROI do *frame* processado o sistema deve ignorar os comandos, a fim de evitar que uma pessoa não autorizada realize comandos sem permissão quando um usuário do sistema está presente. Mediante a testes com o detector do OpenCV para detecção de pessoas não foi possível realizar este controle, pois classificador identifica pessoas de corpo inteiro e como os sujeitos da interação podem estar mais próximos à câmera não foi possível realizar este controle tão facilmente.

Como o sistema propõe o reconhecimento de gestos como meio de interagir com o ambiente e dispositivos, se faz necessário de que mais gestos sejam reconhecidos, sendo necessário o treinamento e disponibilização de

mais comandos para utilização. Para melhorar a qualidade do reconhecimento de gestos define-se como melhoria a troca de segmentação para o uso de *adaptive thresholding*, assim buscando melhores resultados para o modelo de reconhecimento.

Referências

AJIT, Jaokar. **An Introduction to Deep Learning and it's role for IoT/ future cities**. Disponível em: <https://www.datasciencecentral.com/profiles/blogs/an-introduction-to-deep-learning-and-it-s-role-for-iot-future>. Acesso em: 17 de novembro, 2019.

BROWNLEE, Jason. **A Gentle Introduction to Pooling Layers for Convolutional Neural Networks**. Disponível em: <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>. Acesso em: 17 de novembro, 2019.

BROWNLEE, Jason. **What is Deep Learning?**. Disponível em: <https://machinelearningmastery.com/what-is-deep-learning/>. Acesso em: 17 de novembro, 2019.

DALAL, Navneet; TRIGGS, Bill. **Histograms of Oriented Gradients for Human Detection**. Disponível em: <http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>. Acesso em: 9 de novembro, 2019.

DAWSON-HOWE, Kenneth. **A Practical Introduction to Computer Vision with OpenCV**. 1. ED. John Wiley & Sons, 2014.

GARTNER. **Gartner Identifies Top 10 Strategic IoT Technologies and Trends**. 2018. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends>. Acesso em: 15 de novembro, 2019.

GEITGEY, Adam. **Machine Learning is Fun! Part 4:** Modern Face Recognition with Deep Learning. 2016. Disponível em: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>. Acesso em: 9 de novembro, 2019.

GIL, Antonio. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2006.

MATHWORKS. **Image Segmentation & Image Thresholding**. Disponível em: <https://www.mathworks.com/discovery/>. Acesso em: 15 de novembro, 2019.

MORAES, Jairo. **Controle de acesso baseado em biometria facial**. Universidade Federal do Espírito Santo, 2010.

NAKASHIRO, Marta M. **Biometria no Brasil e o registro de identidade civil:** novos rumos para identificação. 2011. 126 f. Tese (Doutorado em Sociologia) – Departamento

de Pós-Graduação em Sociologia, Universidade do Estado de São Paulo, São Paulo, 2011.

NG, Andrew. **What data scientists should know about deep learning**. Disponível em: <https://www.slideshare.net/ExtractConf>. Acesso em: 17 de novembro, 2019.

OPENCV. **OpenCV 2.4.13.7 documentation**. Disponível em: < https://docs.opencv.org/2.4/index.html>. Acesso em: 9 de novembro, 2019.

SAHA, Sparcha. **Hand Gesture Recognition Using Background Ellimination and Convolution Neural Network**. Disponível em: <https://github.com/SparshaSaha/Hand-Gesture-Recognition-Using-Background-Elllimination-and-Convolution-Neural-Network>. Acesso em: 15 de agosto, 2018.

SAHA, Sumit. **A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way**. Disponível em: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Acesso em: 17 de novembro, 2019.

SHALEV-SCHWARTZ, Shai; BEN-DAVID, Shai. **Understanding Machine Learning:** From Theory to Algorithms. 1 ed. Cambridge University Press, 2014.

TOLBA, A. S.; EL-BAZ, A. H.; EL-HARBY, A. A. **Face Recognition:** A Literature Review, International Journal of Computer, Electrical, Automation, Control and Information Engineering. vol. 2, no. 7, 2008.