Samuel Pereira de Godoy
Andrey Cesar Morastico

Segurança física com reconhecimento facial

Trabalho de conclusão de curso

Curitiba
2019

Samuel Pereira de Godoy
Andrey Cesar Morastico

Segurança física com reconhecimento facial

Monografia apresentada para disciplina de trabalho de conclusão de curso de Engenharia de Computação do Centro Universitário UNINTER, como requisito parcial para obtenção do título de Engenheiro de Computação.

Orientador:

Nome do orientador: Charles Fung
E-mail do orientador: charles.f@uninter.com
Assinatura do orientador:

Curitiba
2019

LISTA DE FIGURAS

LISTA DE TABELAS E QUADROS

LISTA DE ABREVIATURAS E SIGLAS

K-NN: K Nearest Neighbor (Vizinho mais próximo)

AdaBoost: Adaptive Boosting. (Reforço adaptável)

ISO: *International Organization for Standardization (*Organização Internacional para Padronização*).*

PCI DSS: *Payment Card Industry Data Security Standard (*Padrão de segurança de dados da indústria de cartões de pagamento*).*

RTSP: *Real Time Streaming Protocol (*Protocolo de transmissão em tempo real*)*

CFTV: Circuito Fechado de TV

RNA: Rede Neural Artificial

DVR: Digital Video Recorder. (Gravador de vídeo digital)

NVR: Network Video Recorder. (Gravador de Vídeo em Rede)

IP: *Internet Protocol. (*Protocolo de internet*)*

TI: *Information Technology*. (Tecnologia da informação)

LISTA DE SÍMBOLOS

$\sum$: Somas definidas em alguma sequência, como uma progressão aritmética.

n : Número de valores da amostra.

$\in$: Pertence

SUMÁRIO

RESUMO

Neste projeto é demonstrada a implementação de um sistema de reconhecimento facial utilizando o algoritmo Viola Jones em conjunto com o AdaBoost para a detecção facial e com classificadores de características do rosto. Após o processo de detecção foi utilizado o algoritmo dos vizinhos mais próximos (K-NN), aplicando a distância euclidiana para a identificação facial, através de comparação de uma imagem fotografada com imagens registradas. O sistema de reconhecimento facial proposto visa atender também parte das regulamentações relacionado a segurança física da ISO 27001:2013, PCI, NIST e LGPD. Os testes realizados demonstram que o sistema possui média de precisão de 57,22% possuindo baixo custo computacional.

Palavras-chave: Reconhecimento facial, Detecção Facial, Segurança Física, Segurança da informação.

ABSTRACT

This project demonstrates the implementation of a facial recognition system using the Viola Jones algorithm in conjunction with AdaBoost for face detection and face feature classifiers. After the detection process, we used the nearest neighbor algorithm (K-NN), applying a Euclidean distance for facial identification, comparing a photographed image with recorded images. The proposed face recognition system is also intended to comply with part of the physical safety regulations of ISO 27001: 2013, PCI, NIST and LGPD. Tests show that the system has an average accuracy of 57.22%, with low computational cost.

Palavras-chave: Face recognition, face detection, physical security, information security.

1. INTRODUÇÃO

Segundo HINTZBERGEN et al. (2015) a segurança física e da informação são antigas, possuem mais de dois milênios de idade. Os egípcios utilizavam hieróglifos desconhecidos para ocultar informação e os chineses construíram fortalezas como por exemplo a grande muralha, por isso a segurança física e a da informação estão juntas presentes na segurança de tecnologia da informação. A segurança física é importante para algumas empresas, fazendo-se necessário a realização de uma avaliação de risco para saber como será a rigidez dos controles de acessos, que são portas eletrônicas, catracas e outros métodos de acessos, podendo ser físicos ou lógicos. para que pessoas não autorizadas acessem informações ou locais da empresa. Os controles de acessos físicos podem ser guardas de segurança e gerenciamento de acesso eletrônico. O controle de acesso com guardas é custoso, mas é necessário em caso de detecção de invasão. O gerenciamento eletrônico pode ser feito com métodos tradicionais como o cartão eletrônico RFID (Identificação por rádio frequência), mas é um método vulnerável, porque o cartão de identificação eletrônico ou não, pode ser falsificado. Situações que envolvem barreiras físicas como portas podem ser protegidas por impressão digital ou íris, mas irá registrar somente a pessoa que liberou o acesso, exigindo um melhoramento de controle. O reconhecimento facial é uma proposta para auxiliar na detecção de invasão, pois os guardas não conseguem reconhecer as pessoas de forma efetiva. Além dos problemas existentes de invasões, os seguranças possuem os fatores humanos como necessidades fisiológicas, memória, randomização por parte de protocolos de segurança, fluxo de pessoas constante e folga/férias dificultando os guardas conhecerem as pessoas autorizadas ou não, facilitando para pessoas mal-intencionada.

2. OBJETIVOS

2.1. Objetivo Geral

O documento demonstra o funcionamento de um sistema de reconhecimento facial assistido, exemplificando os benefícios de quando é operado por guardas de segurança física em ambientes empresariais, sendo que algumas empresas precisam de monitoramentos constantes conforme as regulamentações da ISO 27001:2013, LGPD, NIST e PCI.

2.2. Objetivos Específicos

O projeto deste trabalho propõe-se os seguintes objetivos específicos:

- Construir códigos em linguagem Python;
- Utilizar a biblioteca de visão computacional;
- Trabalhar com técnicas de processamento de imagens;
- Utilizar e explicar algoritmos de detecção de face;

2.3. Problematização

A segurança física em ambientes corporativos obedece às regulamentações de segurança da informação ISO 27001:2013 (Organização Internacional de Normalização) e PCI (Padrão de Segurança de Dados da Indústria de Cartões de Pagamento), são compostas por guardas de segurança, que estão sujeitos a leis trabalhistas. A lei do repouso (BRASIL, 1949), exige que os funcionários tenham o direito de tirar férias e que tenham uma carga horária máxima de trabalho por dia. Sendo assim cada segurança está sujeito a troca de turnos, fazendo com que não memorizem as pessoas autorizadas.

Os guardas de segurança física não estão sujeitos somente a leis trabalhistas, mas aos fatores humanos. Segundo Nitza (E. e R., 2017) todos esquecemos, possuímos uma curva de esquecimento, porque a memória é seletiva e não lembra de tudo.

As regulamentações recomendam que guardas monitorem ambientes corporativos, porém não podem vigiar o tempo todo, além de não lembrarem de

tudo, sendo assim de acordo com HINTZBERGEN et al. (2015) as empresas podem estar sujeitas a invasões.

2.4. Justificativa

O anexo A.11.1 da ISO (*International Organization for Standardization*) 27001:2013 estabelece padrões para gerenciamento de acesso físico, enfatizando a proteção através de controles de entradas em ambientes corporativos para apenas pessoas autorizadas. (ISMS.online, 2019).

A certificação PCI DSS (*Payment Card Industry Data Security Standard)* para empresas que processam cartões de crédito exige o monitoramento de controles de acesso para vigiar entras e saídas. Os visitantes em uma empresa precisam ser monitorados e identificados.

Segundo a lei geral de proteção dos dados (BRASIL,2018) o art. 7º informa que os dados pessoais só podem ser tratados com o titular ciente e para cumprimento de lei ou regulamentação.

O monitoramento constante deve ser feito para atender as regulamentações, sendo necessário atender a lei de proteção dos dados, exigindo que seja detectado e registrado somente pessoas cadastradas, com autorização formalmente. Um sistema de gravação por câmeras não atende as normas, porque não é seletivo, gravando qualquer pessoa, exigindo um desenvolvimento de um sistema que salva os dados apenas de quem permitiu e que está cadastrado.

3. FUNDAMENTAÇÃO

3.1. Detecção Facial

Segundo Cha Zhang e Zhengyou (2010) “Detecção facial é uma técnica de processamento de imagem e visão computacional para determinar a existência, ou não, de faces numa determinada imagem e, caso exista(m), retornar à localização da(s) mesma(s) ”. Uma das várias aplicações da detecção facial, pode-se contar quantas pessoas estão em um determinado ambiente. Em sistema de segurança, o usuário pode programar um horário que não pode possuir pessoa em um determinado ambiente restrito caso aconteça, dispara um alarme ou pode ser alertado diretamente o responsável. Outros processamentos que podem ser realizados, podem-se citar: controle de tráfego em rodovias, obter tamanho de um objeto, detecção de sorrisos em câmeras/ajuste de foco. Para essa detecção é necessário utilizar algoritmos como Viola-Jones, que através de cálculos identifica se possuem faces ou não, em uma determinada imagem. Duas medidas utilizadas para avaliar a qualidade do algoritmo podem ser aplicadas: A primeira é a quantidade de objetos que o algoritmo identificou como face incorretamente, pode-se ser chamado também de falso positivo. O segundo é a quantidade de faces que não foram identificadas, chamado de falso negativo. O algoritmo com melhor qualidade é aquele que possui menor quantidade de valores, tanto para o primeiro caso, quanto para o segundo.

O primeiro método de detecção, de acordo com Omaia (2009, apud Silva,2018), “este foi o primeiro método de detecção de face em tempo real em vídeo, conseguindo processar até 15 quadros por segundo”.

3.1.1. Integral de Imagem

Integral de imagem ou em seu termo em inglês *integral image*, é uma técnica que permite em computação visual, que faça diversos cálculos de maneira eficiente, em uma sub-região da imagem escolhida. Esta técnica pode ser efetuada de maneira significativa de tempo de processamento da imagem. (TOSCANO,2011).

3.1.2. Algoritmo Viola-Jones

Paul Viola e Michael Jones em 2001, citado por BRAGA (2013), propuseram uma abordagem para detecção de objetos em imagens baseado em três conceitos:

- Integral de imagem.
- Treinamento de classificadores usando boosting.
- Uso de classificadores em cascata.

Integral da imagem ou conhecida como tabela de soma de áreas, foi um algoritmo proposto em 1984 por Frank Crow (BRAGA,2013), que avalia eficientemente a soma dos valores dos pixels, por intensidade dos níveis de cinza em uma determinada sub-região de uma área retangular da imagem.

A equação 1 exemplifica como é calculado a integral da imagem em uma determinada coordenada (x,y).

$$ii(x,y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x',y') \qquad (1)$$

$ii(x,y)$ É a somatória da intensidade dos pixels nas coordenadas do pixel (x,y) e i (x',y') é a imagem original.

Pode ser observado, que a coordenada ii(x',y') é a soma dos pixels acima de y e a esquerda de x, mais o próprio ponto (x,y) conforme a figura 1.

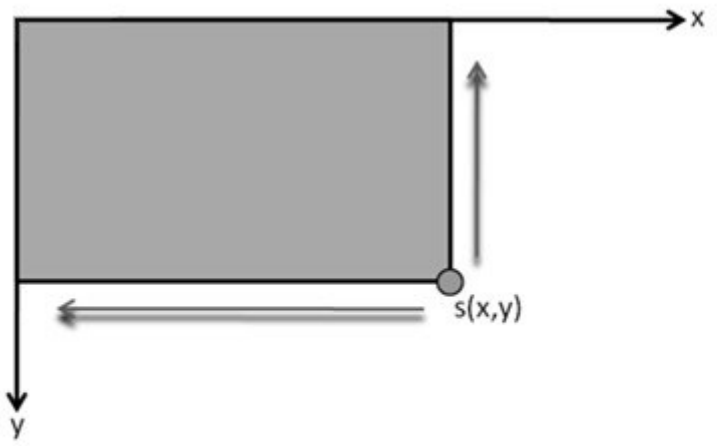

Figura 1 - Somatória da tabela resumida

Fonte: Computer Science Source (2019).

Desta forma, os coeficientes da imagem podem-se ser calculados com apenas uma varredura na imagem original conforme a equação (2):

$$ii(x,y) = i(x,y) + ii(x-1,y) + ii(x,y-1) - ii(x-1,y-1) \qquad (2)$$

Define-se (x,-1)=0 e (-1,y)=0 para contornar os casos nos quais as coordenadas dos pixels estão fora dos limites da imagem. A tabela abaixo, exemplifica esta definição:

imagem original			
5	2	5	2
3	6	3	6
5	2	5	2
3	6	3	6

integral da imagem original				
0	0	0	0	0
0	5	7	12	14
0	8	16	24	32
0	13	23	36	46
0	16	32	48	64

Tabela 1 - Exemplo de cálculo da integral

Fonte: Autoria própria (2019).

A tabela 1 mostra que os números são a intensidade dos pixels, e os valores “0” são as coordenadas que estão fora dos limites da imagem.

deste modo Segundo Silva(2015) o uso da imagem integral acelera os cálculos das características, podendo somar a quantidade de pixels em qualquer área do retângulo, utilizando apenas quatro referencias (A,B,C e D)matricial da tabela integral da imagem, fazendo cálculos de soma e subtração, de acordo com a equação (3):

$$\sum_{(x,y)\in ABCD} i(x,y) = ii(D) + ii(A) - (ii(B) + II(C)) \quad (3)$$

Conforme a equação 3 os termos: ii(A), ii(B), ii(C), ii(D). São os pontos mostrados na tabela 2.

Para exemplificar o uso desta metodologia, pode calcular uma sub-região da integral da tabela 1 separa-se A, B, C, D conforme a tabela 2.

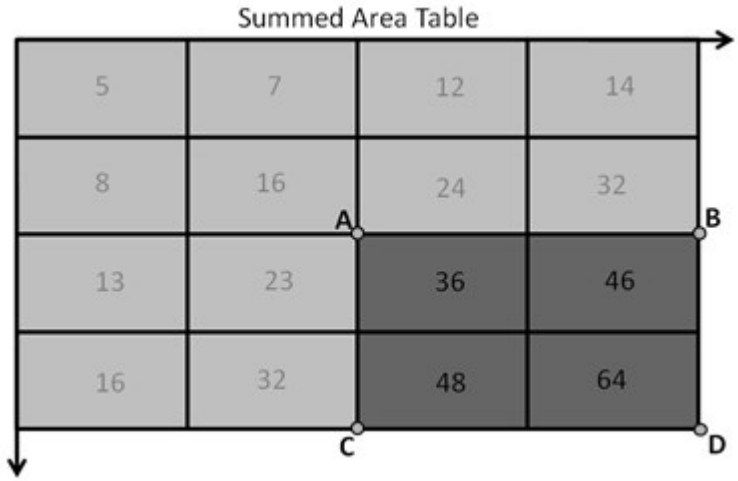

Tabela 2 - Área pretendida

Fonte: Computer Science Source (2019).

Pode-se observar que as somas dos pixels A é 16, B é 32, C é 32 e D o próprio 64. Após a identificação de ABCD basta utilizar a equação 3, a qual resultará em 16. Retornando a imagem original e fazendo a soma da subárea nota-se que o valor é o mesmo. Após esse processo pode-se identificar um padrão utilizando a características Haar-*like*, que são máscaras que possuem uma região preta e outra branca. Pode-se subtrair uma da outra, com os resultados, pode-se utilizar os valores dos pixels (diferença de intensidade luminosa entre áreas da imagem), para a detecção por exemplo dos olhos. A partir da Figura 2 pode ser observado quatro tipos de características que podem

ser usadas. Para o cálculo são necessárias oito consultas na tabela da imagem integral, conforme a primeira Haar-*like* marcado na figura2A.

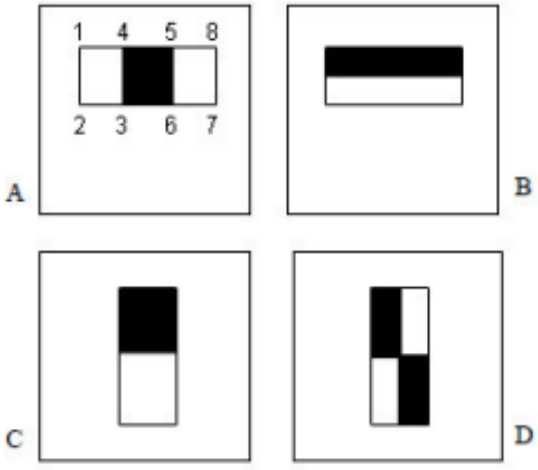

Figura 2 - Características de extração

Fonte: Luiz Filipe Zenicola Braga (2019).

Cada uma da Haar-*like* tem melhor detecção para determinada área que deseja detectar um padrão, por exemplo a figura 2(B) pode identificar diferenças de intensidade entre a parte superior e a parte inferior de uma região. A figura 2(B) pode ser utilizada para identificação dos olhos onde a sobrancelha é mais escura que a região dos olhos (Braga, 2019). Existem diversos padrões para identificar uma área específica, sendo uma melhor busca é feito uma combinação de diversas Haar-*like* quanto maior o número, melhor é a detecção. De acordo com Braga (2019), existe também combinações para identificar o que não são faces. A resolução base das máscaras utilizada no algoritmo é a de 24x24 pixels.

Para a detecção de objetos no algoritmo de Viola-Jones, o segundo passo é o treinamento de classificadores. Que irá receber um conjunto de características de Haar-*like* e treiná-lo como imagens positivas (faces) e imagens negativas (não faces).

É necessário um algoritmo que aprenda funções de classificação, como algoritmo Naive bayes, redução de dimensional, algoritmo AdaBoost (*Adaptative Boosting*).

O utilizado neste projeto foi o Boosting, que utiliza vários classificadores fracos combinados para encontrar um classificador preciso e com melhor índice

de acerto. Cada um dos classificadores fracos é necessário uma precisão média com uma porcentagem de acerto de no mínimo 51%.

AdaBoost é basicamente um algoritmo que tem como objetivo construir um classificador forte a partir de combinações lineares de vários classificadores fracos. Conforme a equação abaixo representa o funcionamento desse algoritmo:

$$f(x) = \sum_{t=1}^{T} a_t h_t(x) \tag{4}$$

h_t Representa classificadores fracos, que pode assumir valores 0 ou 1 para exemplos negativos e positivos.

x Representa a janela respectivamente 24x24.

a_t É o peso do classificador.

Um classificador fraco pode ser expresso pela função da característica (f), de um threshold (θ) e de uma polaridade (p) para indicar a direção da igualdade, como a seguir a equação:

$$h(x, f, \theta, p) = \begin{cases} 1, & se\ pf(x) < p\theta \\ 0, & caso\ contrário \end{cases} \tag{5}$$

É possível fazer o cálculo de um classificador forte, que está representado pela equação abaixo:

$$H(x) = \begin{cases} 1, & f(x) \geq \frac{1}{2}\sum a_t \\ 0, & caso\ contrário \end{cases} \tag{6}$$

O algoritmo AdaBoost, usa as fórmulas tanto para as características das Haar-*like* são mais adequadas, como também utiliza para treinar os classificadores com essas características escolhidas. Pode-se observar as características escolhida pelo AdaBoost na imagem abaixo:

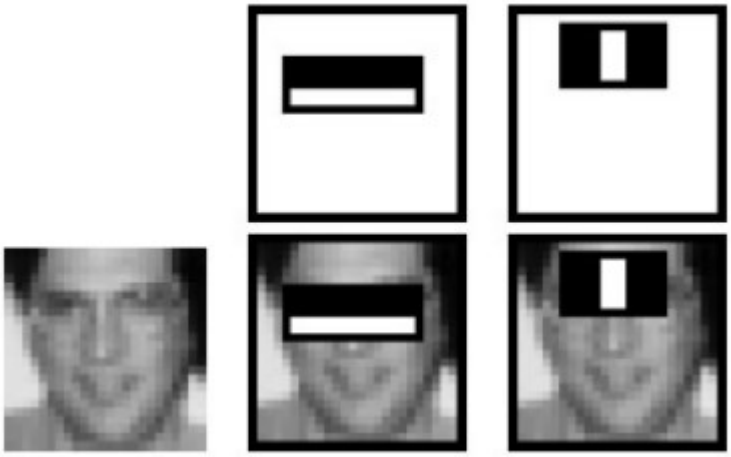

Figura 3 - Características selecionadas pelo algoritmo AdaBoost

Fonte: BRAGA (2019).

A Próxima etapa é combinar classificadores fortes em modo de cascata efetivando processamentos em regiões da imagem em busca de padrões. Para cada fase do modo em cascata e aplicado um classificador de melhor precisão e complexidade do anterior, de modo que as características que não está sendo procurada na imagem seja rejeitada, evitando que as próximas fases sejam executadas desnecessariamente. Com isso os que não são faces são rejeitados de início na primeira fase e o que são faces são analisados como maior precisão. A figura 4 mostra o funcionamento do processo de classificação em cascata, com o início da imagem e as fases como C1, C2, C3 e sucessivamente. Caso seja diferente das características procuradas. Rejeita a entrada, se não aceita a entrada.

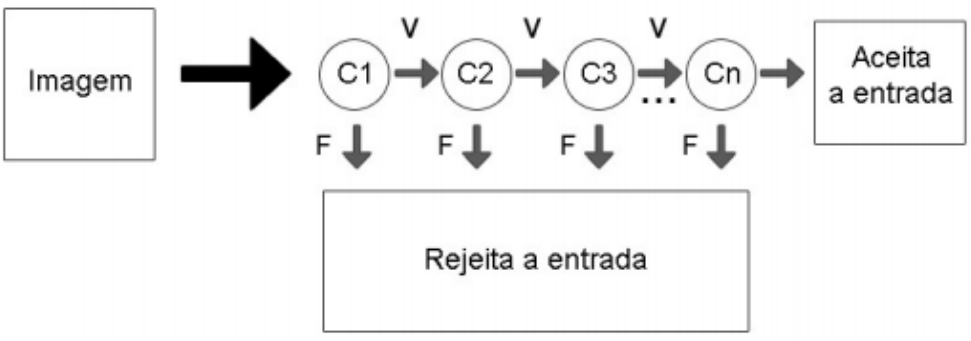

Figura 4 - Cascata de classificadores

Fonte: Luiz Filipe Zenicola Braga (2019).

Segundo Braga (2019) O algoritmo de Viola-Jones, se possuir um bom treinamento, tem uma boa precisão, conseguindo detectar a maioria das faces que são capturadas frontalmente, e uma pequena quantidade de faces. Uma das desvantagens desse algoritmo é a detecção facial de perfil, ou quando apresenta barreiras em frente dos olhos como por exemplo óculos, também quando existe pouca iluminação no ambiente. É possível também fazer detecção em múltiplas escalas mudando. A imagem abaixo mostra como é feito essa varredura, das janelas de detecções buscando padrões definidos pelas características do tipo Haar-*like*, em cada região são aplicados inúmeros classificadores, e a detecção só é válida como positivo se para todos os classificadores da cascata aceite a entrada (C1 da figura 4).

Figura 5 - Extraindo características

Fonte: BRAGA (2019).

3.2. Reconhecimento Facial

“O reconhecimento facial é um aplicativo de software biométrico capaz de identificar ou verificar exclusivamente uma pessoa, comparando e analisando padrões com base nos contornos faciais da pessoa.” (Techopedia, 2019). As aplicações para o reconhecimento facial frequentemente utilizado é o desbloqueio do celular, posicionando o celular de frente com o rosto, cursos online também utilizam essa plataforma para confirmar a identidade do aluno.

3.2.1. Sobre o Reconhecimento Facial

“O rosto humano, apesar das variações de pessoa para pessoa, possui uma composição básica que não se altera, lida pelos aplicativos como pontos em comum, que variam de acordo com a complexidade do sistema.” (KLEINA, 2011).

O rosto de um ser humano possui uma composição básica que não se altera, como olhos, boca nariz. Com isso pode ser feita a detecção da face com formas geométricas e logarítmicas e depois montá-lo. Para identificar um rosto é necessária uma câmera para capturar essa foto, podendo identificar alguns pontos em comum, como os dois olhos, a distância entre eles, o nariz, seu comprimento, a boca, as bochechas e o queixo, limitando assim o formato da face e o espaço ocupado por ela. Abaixo uma demonstração do processo.

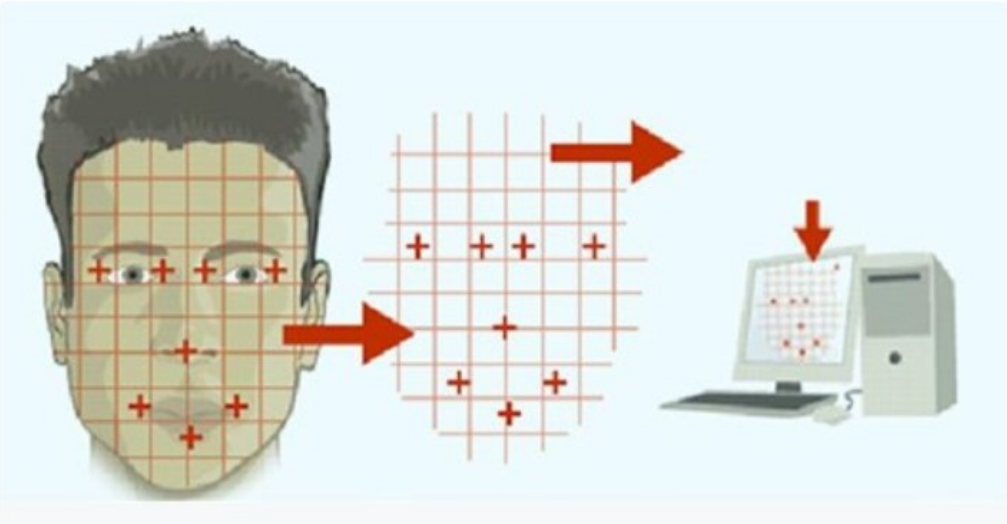

Figura 6 - Formato de pontos gravados.

Fonte: Face Recognition Solution (2019).

Cada ponto é gravado e armazenado no formato de algoritmo em um banco de dados, que através de cálculos é reconhecido.

Uma das principais falhas e limitações do reconhecimento facial, eram mudanças bruscas que poderia afetar a detecção como pouca iluminação ou muita, uso de acessórios e perfil de diferentes ângulos dificultando a detecção, uma forma de ter uma detecção era com o rosto de frente com a câmera e com a iluminação correta.

Com o avanço da tecnologia esses problemas começaram a diminuir, pois começou a surgir ferramentas avançadas própria para reconhecimento fácil com o tamanho e o acesso a um banco de dados gigantesco até mesmo hoje é possível fazer uma detecção facial em 3D que antes era feito a identificação a partir de pontos em um plano 2D. Capturar o formato da cabeça do usuário, o que antes era de forma diferente. Com esse novo formato de captura é possível identificar o ângulo do rosto em diversas posições em relação a câmera, mudanças no ambiente também não afetam mais o reconhecimento.

O reconhecimento Facial é aplicado em diversas situações como de detecção de rostos, melhorar qualidade do retrato. Para que seja feito um reconhecimento necessita primeiramente passando pela fase de detecção, conforme explicado no algoritmo de viola-Jones, já para o reconhecimento facial precisa reconhecer os padrões com K-NN (*K-nearest neighbors*).

3.2.2. Reconhecimento Com K-NN

No algoritmo AdaBoost foi o processo de detecção de face, que em sua saída é um conjunto numérico, que pode ser denominado como vetor característico. A última etapa consiste em classificação dos vetores característicos, chamado como reconhecimento de padrões.

Segundo COSTA (2018), possuem várias técnicas utilizadas para agrupar vetores característicos como classe. Backes e Junior (2016) agrupa essa técnica em 4 classes sendo elas Classificadores elementares, Classificadores Bayesianos, Agrupamento e Redes Neurais Artificiais (RNA).

- **Classificadores elementares -** é baseado em métricas de distâncias. As principais são K-vizinho mais próximo (K-NN) e classificador de protótipo mais próximo.

- **Classificadores Bayesianos –** são classificadores estatísticos, baseado na fundamentação do teorema de Bayes, que classificam um objeto em uma determinada classe C, baseando-se na probabilidade do objeto pertencer à classe C. Nesta categoria possuem a análise linear

discriminante (LDA), análise quadrática discriminante (QDA) e classificadores Naive Bayer.

- **Agrupamento –** utilizado em aprendizado não supervisionado, os algoritmos de agrupamento dividem um conjunto de objetos em agrupamentos, esses objetos são descritos e agrupados utilizando um conjunto de atributos e valores. Como utiliza o aprendizado não supervisionado não existe nenhuma informação sobre a classe ou categoria dos objetos. Sendo seu objetivo colocar os objetos similares em um mesmo grupo e objetos que não são similares em grupos diferentes.
- **Redes Neurais Artificiais (RNA)** – começou com três principais publicações: sendo elas de McCulloch e Pitts (1943), Hebb (1949), e Rosemblatt (1958). Essas publicações mostraram o primeiro modelo de redes neurais simulando “máquinas”, o modelo básico de rede de auto-organização, e o modelo perceptron de aprendizado supervisionado (Fernando,2019). Estas redes são técnicas computacionais que seguem um modelo matemático, orientado na estrutura neural que adquirem conhecimento através da experiência. Podendo conter essa rede neural centenas ou milhares de unidades de processamento. Estes processamentos existem várias unidades, onde essas unidades são geralmente conectadas por canais de comunicação que estão associados a certo peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. A sua operação de uma unidade de processamento pode ser resumida por: Sinais apresentados na entrada; cada sinal é multiplicado por um número, ou geralmente peso, que indica a sua influência na saída da unidade; depois feito a soma ponderada dos sinais que produz um nível de atividade; se exceder um certo limite (*threshold*) do nível de atividade, a unidade produz uma determinada resposta de saída.

Conforme mostrado as técnicas para classificar, considerando que as características da face possuem classes bem definidas, caso seja separado um conjunto de imagens da mesma face esse conjunto de dados define a classe que representa a face treinada no algoritmo, podendo assim observar

que a abordagem de clusterização não eficaz, sendo que para agrupamento de dados para definir uma classe não é necessário. Como os dados possuem uma variabilidade aleatória, o classificador de Bayer não é recomendável.

Sendo uma das características dos classificadores elementares além de se basear em métricas de distância, eles são lineares, que será fundamental pois o ideal para o treinamento facial é aplicar um conjunto de dados que representa comportamento linear. Segundo COSTA (2018) “pode ser generalizada para classificar conjuntos com distribuição espacial não linear” sendo que as saídas de uma rede neural dependem dos pesos, que não estão inter-relacionados necessariamente, podendo ser visto em (RUSSEL,2010).

3.3. K-Nearest Neighbors

Segundo SILVA (2015), o *K-Nearest Neighbors* (K-NN) é usado em problemas de classificação, inclusive em reconhecimento facial (JOSE; POORNIMA; KUMAR,2012). Este algoritmo se baseia em atribuir a uma amostra desconhecida a classe das K amostras que estejam mais próximas, utilizando métricas de distâncias como:

- **Euclidiana –** Segundo CLÉSIO (2016)” A distância Euclidiana é definida como a soma da raiz quadrada da diferença entre x e y em suas respectivas dimensões”.
- **Mahalanobis –** “Já a Distância Manhattan tem uma definição mais simples na qual é apenas a soma das diferenças entre x e y em cada dimensão.” (CLÉSIO,2016).

Como diversas outras distâncias. De maneira genérica pode utilizar a distância de Minkowski ou normal Lp, definido como:

$$L_p(x_j, x_q) = \left(\sum_i |x_{j,i} - x_{q,i}|\right)^{1/P} \quad (7)$$

Onde x_j é a amostra não classificada e x_q o conjunto de treinamento. Com P = 2 temos a distância euclidiana, para P = 1 distância de Manhattan.

A distância euclidiana é usada para medir a altura, largura ou profundidade, e a de Manhattan é utilizada se elas não são similares, como peso, gênero ou paciente.

Após encontrar as distâncias desconhecidas de um objeto em relação ao conjunto de treinamento, a classe dominante entre as K amostras será determinado como a classe do objeto desconhecido. Caso esses objetos aconteça um empate um critério de decisão pode ser adotado. Uma das alternativas é que a amostra mais próxima do objeto decida a classe.

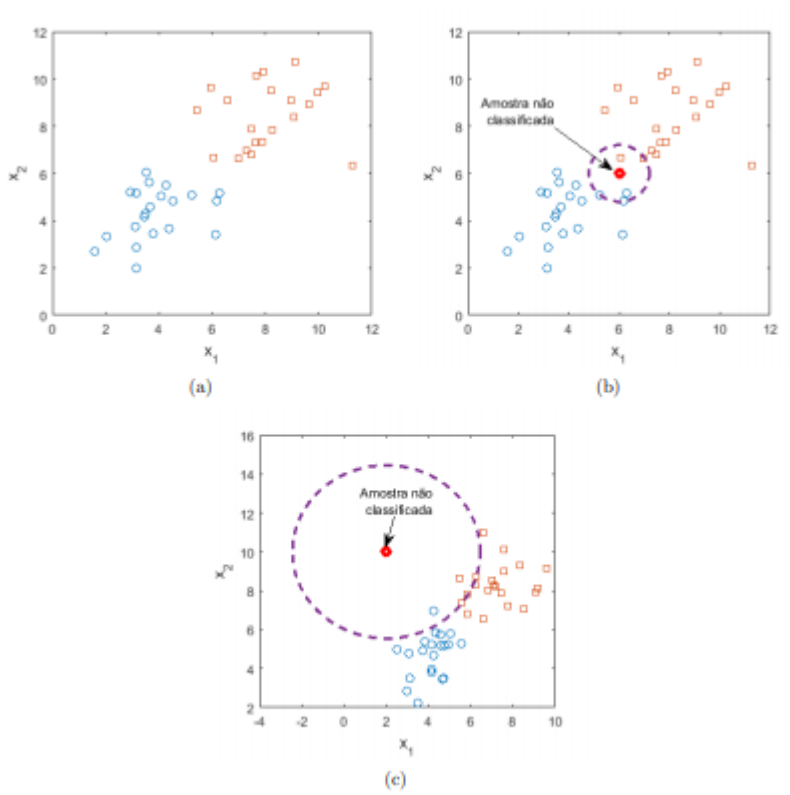

Figura 7 Amostra de dois Grupo dos vizinhos mais próximos

Fonte: COSTA (2018).

Conforme a figura 7 é mostrado um exemplo do funcionamento do algoritmo K-NN a um conjunto de dados x, contendo duas componentes uma x1 e x2. Na figura 7(a) existe no plano cartesiano duas características distinta que representam objetos conhecidos, os pontos circulares azuis pertencem a classe A e os pontos quadráticos laranjas a classe B. Considerando uma amostra não classificada conforme a figura 7(b) localizada no plano cartesiano

x = (6,6), para K = 5, a classe da amostra será aquela que tiver os maiores números de exemplos no conjunto de treinamento (dentro do círculo roxo) dentre os cincos vizinhos mais próximos. Observando a figura 7(b) observamos que os vizinhos mais próximos da amostra desconhecida são os pontos quadráticos azuis logo pertence à classe B. já na figura 7(c) a amostra desconhecida está longe de ambos os grupos classificadores mesmo assim o algoritmo K-NN tenta atribuir a classe B.

O algoritmo K-NN para os problemas em é que a amostra desconhecida sempre será associada a alguma classe, mesmo que os vizinhos K mais próximos estejam longe da amostra.

Segundo COSTA (2018) para garantir bons resultados com o algoritmo K-NN é necessário fazer vários conjuntos de treinamento para obter várias amostras.

3.4. Reconhecimento Facial e Segurança

Segundo a empresa DINO (2019) em sua matéria *Reconhecimento facial revoluciona controle de acesso e segurança predial*, diz que “a demanda pela tecnologia é cada vez maior no país”.

No Brasil o uso de biometria possui um avanço exponencial para diversas finalidades. Em ambientes de grande circulação ou em ambientes que necessitam ser controlados, o reconhecimento facial está sendo uma verdadeira revolução. Pois disponibiliza uma operação inteligente e automática, sendo capaz de reconhecer rostos e aumentando a segurança.

O reconhecimento facial se divide entre duas identificações colaborativas e não colaborativas, está sendo utilizada nos sistemas de CFTV (sistemas de vídeo vigilância) e para o controle de acesso. O software para analisar as imagens que identifica os rostos, os mapeiam e criam códigos para cada pessoa, restringindo com segurança o acesso a operações e perímetros sensíveis.

O reconhecimento facial colaborativo, é onde o indivíduo de forma espontânea autoriza que a câmera tenha seus dados pessoas e faça a leitura do rosto, o mesmo está disposto a posicionar a distância adequada, isto possibilita a liberação

de pessoas autorizadas em ambientes controlados, sem a necessidade de cartões e códigos, facilitando o processo de forma ágil e sem risco de fraudes.

CFTV vem da sigla circuito fechado de TV, consiste em um sistema de monitoramento interno através de câmeras, que fazem gravações para fazer registros e controle em um ambiente corporativo unidos através de um sistema central. Esse sistema consiste em dois tipos o analógico e o digital. Com os sistemas analógicos são conectados aos dispositivos centrais (DVR), por cabos coaxiais exibidos em monitores específicos diferente do sistema digital que se destaca as câmeras IP, a qualidade é superior ao analógico e uma grande diferença que pode ser utilizado por (NVR) que disponibilizar em monitores de alta definição, em celulares, tablets de maneira remota e possibilitando em tempo real utilizando protocolo RTSP.

3.4.1. Protocolo RTSP

Este protocolo RTSP vem da sigla “Real Time Streaming Protocol” em sua tradução é protocolo de fluxo em tempo real, segundo a empresa aprenda CFTV sua principal função é enviar áudio ou vídeo ao vivo de um dispositivo a outro. Os fabricantes de equipamentos de segurança pensaram em uma maneira de por exemplo uma câmera IP e um gravador de diferentes fabricantes conseguisse uma comunicação utilizando assim um protocolo universal podendo assim ser compatíveis com outros dispositivos de diferentes marcas e modelos disponíveis no mercado. (DINO, 2019)

3.4.2. Segurança Física

Segurança física é a parte da segurança da informação, sendo que todas as partes ativas do negócio devem ser fisicamente protegidos.

A segurança física é realizada por gerentes de serviços gerais e técnicos que utiliza técnicas e métodos para determinar a segurança física. Onde que em grandes empresas é fundamental possuir encarregados tanto para segurança física quanto para segurança da informação. A segurança busca uma combinação de medidas organizacionais, estruturais e eletrônicas. Sendo a capacidade de detectar um intruso em ambientes não autorizado, também a

detecção é fundamental para reduzir os mínimos riscos de qualquer dano. Podendo ser colocados sensores automáticos para que se houver uma detecção não autorizada seja relatado a um segurança responsável pelo monitoramento. Todos os ativos do negócio devem ser protegidos de ameaças e riscos a esses ativos, as medidas de segurança física são tomadas para proteger a informação contra furtos, acesso não autorizado e vandalismo.

Uma forma de proteger escritórios, salas e instalações importantes, é manter pessoas que não são autorizadas fora do alcance desses locais, dificultando caso haja a tentativa por controles de acessos, dificultando para invasores terem acesso a esse local. Deve ser sempre aplicado um conjunto com outras medidas, não apenas com seguranças, equipamentos eletrônicos mais reduzir qualquer informação do local. Quando locais são utilizados para processamento de dados confidenciais, é importante garantir que não seja possível espionar de fora do local.

3.5. Controle De Acesso

O dono do negócio é a pessoa responsável por um os demais processos, subprocessos ou atividades. Essa responsabilidade inclui a administração de quem possui a liberação para ter acesso à aos ativos do local, incluindo ativos de informação. Os requisitos podem vir de objetivos de negócios, legais e outros requisitos regulatórios. É necessário fazer uma análise de risco do local, para obter o quão restritos esses controles de acesso devem ser, procurando eliminar a possibilidade de riscos a obtenção de acesso a ativos. Os controles de acesso são uma combinação de controles de acesso lógicos e controles de acesso físico.

Segundo Hintzbergen *et al*. (2018, p.83) “Uma política de controle de acesso deve ser estabelecida, documentada e revisada com base nos requisitos de negócio e de segurança da informação” (ISO 27002:2013 definição da seção 9.1.1). Com isso evita de uma pessoa não autorizada ganhe acesso lógico a qualquer área, que tenha valor para a organização. Normalmente em organizações os acessos são responsabilidade de um gerente. Autorização consiste em um conjunto de permissões, permissões mais simples que são por exemplo uma o direito de ler um determinado documento ou alterá-lo as permissões pode ser

complexa que pode ser considerada também como pagamento de contas bancárias. Alguns exemplos de tipos de acessos que necessariamente devem ser considerados para controles de acesso:

- Acesso a redes e serviços de rede;
- Acesso a aplicações de negócio;
- Acesso a equipamentos de TI;
- Acesso à informação;

3.6. Gestão De Acesso Do Usuário

A gestão de acesso do usuário busca prevenir que pessoas não autorizadas tenham acesso a ativos da organização, que sejam acessados apenas por usuários autorizados, para isso necessita as seguintes características:

- Registro e cancelamento de registro de usuário;
- Provisionamento de acesso de usuário;
- Gestão de direitos de acesso privilegiado;
- Gestão de informações secretas de autenticação de usuários;
- Revisão dos direitos de acesso de usuário;
- Remoção ou ajuste dos direitos de acesso;

Para os mesmos envolve uma série de etapas como identificação do usuário, a autenticação deste usuário e a autorização do usuário para o acessar um ativo. Na identificação por exemplo apresenta o cadastro do usuário por exemplo com seu reconhecimento facial. Então o sistema deve conferir se o rosto está cadastrado no sistema e se não possui nenhuma a data vencida, em ambos os testes forem válidos, o usuário será autenticado.

3.7. Lei Geral De Proteção De Dados

A lei geral de proteção de dados pessoais LGPD foi sancionada por Michel temer em 2018 que entrará em vigor em 2020. Esta lei foi inspirada na *General Data Protection Regulation* (GDPR), da União Europeia, a LGPD definindo a

respeito de vários aspectos sendo eles dados pessoas, dados pessoais sensíveis, controle, processamento, consentimento, anonimização, entre outras. Os dados pessoais são as informações relacionadas a pessoa natural identificada ou identificável, podemos citar como CPF, endereço residencial e RG. Os dados de uma pessoa jurídica, como CNPJ, razão social não são considerados como dados pessoais. Dados pessoais sensíveis são dados que podem gerar de uma pessoa discriminação, afetando sobre sua origem racial ou étnica, convicção, religião, opinião política, dado genético ou biométrico. A lei objetiva a preservação da privacidade e da liberdade, garantindo ao usuário o controle sobre seus dados, evitando mal-uso pela parte de terceiros. A lei também se aplica quando a empresa pode utilizar esses dados tanto para armazenar, processar e transferir esses dados. (BRASIL, 2018)

3.7.1. Penalidade por não cumprir a Lei

Caso haja descumprimento da Lei LGPD podem envolver proibições total ou parcial de atividades relacionadas a tratamento de dados. A empresa que não tiver conformidade com a lei terá multa correspondente até 2% do faturamento da empresa ou conglomerado limitado até cinquenta milhões por infração cometida. (BRASIL, 2018)

3.8. Instituto Nacional De Padrões e Tecnologia

Instituto nacional de padrões e tecnologia (NIST) foi fundada em 1901 fazendo parte do departamento de comércio dos EUA. O NIST é um dos mais antigos laboratórios de ciências físicas do país. Sendo responsável energia elétrica inteligente, registros eletrônicos de saúde, relógios atômicos, nano materiais. Grande parte de produtos e serviços depende da tecnologia, que são fornecidos pela NIST. “Hoje, as medições do NIST suportam a menor das tecnologias às maiores e mais complexas criações feitas pelo homem” (NIST,2015).

3.9. NIST 800-53

O NIST 800-53 conhecido também como publicação especiais NIST 800-53, é um conjunto de padrões e diretrizes para ajudar agências e contratados federais a atender aos requisitos estabelecidos pela lei federal de gerenciamento de segurança da informação (FISMA). NIST para seu controle se divide em três classes com base no impacto – baixo, moderado e alto. Se dividindo em dezoito famílias diferentes. A família de controle de segurança NIST SP 800-53 são:

- Controle de acesso
- Auditoria e prestação de contas
- Sensibilização e Formação
- Gerenciamento de configurações
- Planejamento de contingência
- Identificação e autenticação
- Resposta a Incidentes
- Manutenção
- Proteção de mídia
- Segurança de Pessoal
- Proteção Física e Ambiental
- Planejamento
- Gerenciamento de Programas
- Avaliação de risco
- Avaliação e Autorização de Segurança
- Proteção de sistemas e comunicações
- Integridade de sistemas e informações
- Aquisição de Sistemas e Serviços

A NIST 800-53 traz benefícios como melhoria da segurança dos sistemas da informação da organização, fornecendo uma parte fundamental para o desenvolvimento de uma infraestrutura organizacional segura.

Uma das práticas recomendadas do NIST SP 800-53 é analisar, educar e avaliar. Educar os funcionários sobre as etapas executar para se tornarem

compatíveis com o NIST, existe um número de controle de gerenciamento estabelecidos pela NIST 800-53 que a equipe de gerenciamento deve seguir. A outra parte avaliar necessariamente é preciso medir as políticas e processos de segurança, implementando ferramentas que fornecem métodos para medir e avaliar processos de segurança. Analisar é o entendimento, precisando entender quais são as ameaças para os dados de informação. As soluções líderes para análise e proteção dos dados regulamentados são como PCI.

3.10. Padrões De Segurança De Dados De Industria De Pagamento Com Cartão

Padrões de segurança de dados de indústria de pagamento com cartão ou conhecido como Payment Card Industry – Data Security Standard (PCI-DSS) Segundo Sewall (2017) “é um conjunto de requisitos para a proteção de dados de cartões de pagamento.”

Estes padrões são recomendados para pessoas ou empresas, que necessitam trabalhar com sistemas de cartões que é preciso ser armazenado, transmitido e processado.

O PCI é expresso por certos requisitos como:

- Implementando senhas exclusivas
- Criptografia de dados do titular do cartão
- Verificações de integridade do sistema de rotina
- Monitorando o acesso à rede
- Aplicação de políticas de segurança da informação

As categorias são compostas dividida por doze seções, cada seção possui seu objetivo próprio em um subconjunto de itens de ação. A maioria dos requisitos de conformidade com PCI é atendida virtualmente. Uma das sessões dedicada a restrições do acesso físico aos dados do cartão, é o requisito nove que envolve a segurança física.

3.11. Requisito 9: Segurança Física

O primeiro requisito 9.1 é a recomendação que os sistemas para possuírem segurança, “é necessário usarem câmera de vídeo ou mecanismos de controle de

acesso para monitorar o acesso físico individual a áreas sensíveis” (Sewall,2017). Sendo que qualquer área que armazene dados do titular do cartão ou processe deve ser restrito ao acesso físico. Esse sistema deve ser confiável visando ser resistente a alterações ou violação, sendo também capaz de gravar imagens por um período de tempo definido cerca de no mínimo três meses. Garantindo que a equipe de segurança tenha acesso ao histórico de imagens, caso aconteça uma violação dos dados.

3.12. Engenharia Social

Hoje em dia pessoas mal-intencionadas com conhecimento técnicos, utiliza para se infiltrar, procurando falhas de sistemas computacionais como também computadores desprotegidos, obtendo dados confidenciais. Com esses ataques por invasores é necessário investimento de novas tecnologias para reforçar as defesas tanto do computador quanto da rede. Mesmo havendo ferramentas e soluções contra-ataques aos dados pessoas, existe métodos de invasões de diferentes táticas. Uma delas são chamadas de Engenharia Sociais, que busca explorar falhas encontrada nas organizações, sendo a principal delas a psicologia humana. Esta tática é onde induz uma pessoa a entregar os acessos confidenciais da organização comprometendo o sistema. Engenharia Social é o termo que abrange um amplo espectro de atividades maliciosos. As cincos principais atividades usadas são *phishing*, pretexto, isca, *quid pro quo* e utilização não autorizada. (BISSON ,2019)

3.12.1. Ataque Do Tipo *Phishing*

PHISHING é um tipo de ataque que se baseia em 3 princípios

- Obtenha informações pessoais, como nomes, endereços e números de seguridade social.
- Use links reduzidos ou enganosos que redirecionam os usuários para sites suspeitos que hospedam páginas de destino de *phishing*.
- Incorpore ameaças, medo e um senso de urgência na tentativa de manipular o usuário para responder rapidamente.

Esses ataques *phishing* existe pelo menos seis subcategorias diferentes de ataques. Porém esse tipo de ataque muitas vezes são criados com erros como de ortografia e gramática. Mesmo obtendo esses erros o objetivo é roubar credenciais como a obtenção do login do usuário e outros dados pessoais.

3.12.2. Pretexto

Este tipo de ataque concentra em criar um bom pretexto, ou um cenário projetado para ser usados para tentar roubar as informações pessoais de suas vítimas. A pessoa mal-intencionada convence a vítima a fornecer certas informações do alvo para confirmar sua identidade. Com esses dados pessoas eles usam para cometer roubos de identificação ou usar para ataques secundários. Esses ataques um deles é conversar com a vítima a fazer algo através das fraquezas digitais ou física de uma organização. O invasor pode passar como um funcionário de serviços de TI, se infiltrando na equipe de segurança física de uma empresa-alvo para entrar no prédio. Sendo a diferença entre os ataques de *phishing*, utiliza o medo e a urgência em proveito próprio, enquanto o pretexto necessariamente é a construção de um falso senso de confiança com a vítima. Diversos ataques buscam se disfarçar como pessoa de RH ou funcionários no desenvolvimento financeiro. Segundo BISSON (2019) diz que “Esses disfarces permitem atingir executivos de nível C, como a Verizon encontrou em seu Relatório de investigações de violação de dados de 2019 (DBIR)”.

3.12.3. ISCA

Esse tipo de ataque é semelhante a ataques de *phishing*. Sua diferença é a promessa de um produto utilizado para atrair as vítimas. Podendo aproveitar a downloads gratuitos como de músicas filmes, para induzir a vítima a entregar suas credenciais de login. O ataque isca não é usado apenas para ataques online, mas também por curiosidade humana, por meio do uso de mídias físicas.

Segundo BISSON (2019) em um acontecimento desse tipo de ataque em 2018, KrebsOnSecurity noticiou uma campanha de ataques contra-ataques

governamentais estaduais e locais nos Estados Unidos. Onde que a operação enviou envelopes carimbados na China com uma carta confusa e junto um CD. Que despertou a curiosidade dos destinatários, e assim ao executar o CD seus computadores eram afetados por malware.

3.12.4. QUID PRO QUO

Este tipo de ataque é semelhante a isca, que promete a vítima um tipo de benefício a troca de informação. Este fornece em forma de um serviço, enquanto o ataque do tipo ISCA assume a forma de um produto. Os tipos mais comuns de ataques quid pro quo, são fraudadores que se passa por administradores de segurança social entrando em contato com indivíduos aleatórios, informando que houve uma falha ou problema no computador, e após o ocorrido pede que confirmem seu número de seguro social, tendo assim o roubo de informações pessoais. (BISSON ,2019)

3.12.5. Utilização Não Autorizada

Um dos tipos de ataques não autorizado ou chamado “pegar carona”. O invasor sem autorização segue o funcionário para a área restrita. Podendo se passar como um motorista ou algum auxiliar, após a autorização do funcionário e abrir a porta o invasor pede para segurar a porta, e ter acesso ao prédio.

Este tipo de ataque “não autorizado” não funciona em todas as corporativas, como grandes empresas que possuem cartões chaves. Porém em organizações de porte médio a pessoa mal-intencionada normalmente inicia com conversas com o intuito de passar pela recepção. (BISSON ,2019)

4. METODOLOGIA

Este projeto é exploratório, visando demonstrar a utilidade do reconhecimento facial no cenário composto de legislações de proteção de dados pessoais e trazendo a ideia de transformar o reconhecimento facial como uma ferramenta para vigilantes ou porteiros.

O desenvolvimento do projeto exigiu 6 etapas para ser executado:

1. Analisar as legislações vigentes: estudo de legislações e impactos de o desenvolvimento de um sistema computacional composto por dados de pessoas com autorização;
2. Estudar segurança da informação: Identificação de padronizações referentes a segurança da informação;
3. Estudar tipos de detecção facial: o rosto é algo complexo para ser detectado, exigindo o estudo de diversos algoritmos;
4. Estudar forma básica de classificação de imagem;
5. Implementar um protótipo: a proposta exigiu a implementação para que fosse feito uma prova de conceito do reconhecimento facial por classificação de imagem;
6. Avaliar: o sistema exige uma avaliação para que saibamos riscos evidentes;

Antes de iniciar o processo de reconhecimento, faz-se necessário salvar rostos para que o sistema tenha conhecimento para poder comparar com as detecções. Este processo é demonstrado na figura 8.

Figura 8 - processo de salvar autorizados

Fonte: Autoria própria (2019).

Após o processo de salvar os rostos que serão reconhecidos ser finalizado, o procedimento de reconhecimento pode ser iniciado como demonstrado na figura 9.

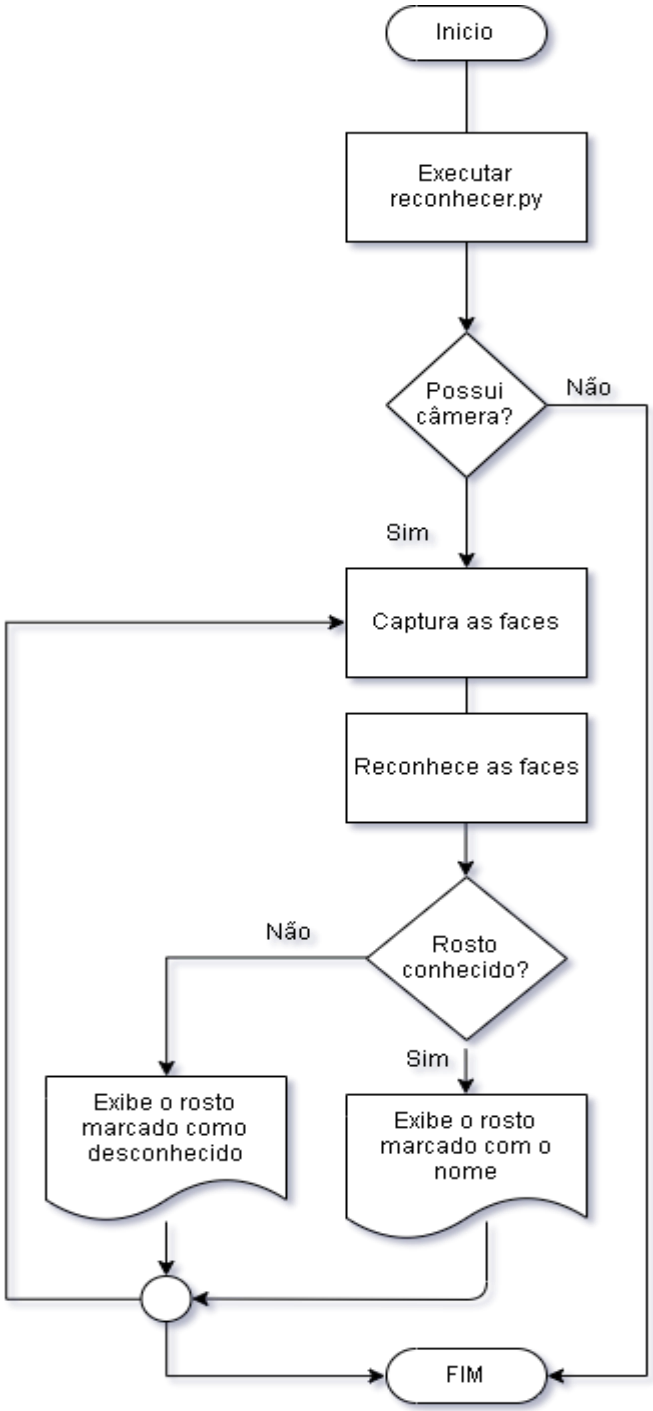

Figura 9 - Processo de reconhecimento

Fonte: Autoria própria (2019).

A figura 10, mostra o diagrama de sequência, que é executado primeiro o programa dataset.py que irá tirar uma sequência de foto para identificar uma

face. Após o registro, executa o programa Reconhecimento.py onde se tiver registrado o rosto irá aparecer o nome da pessoa.

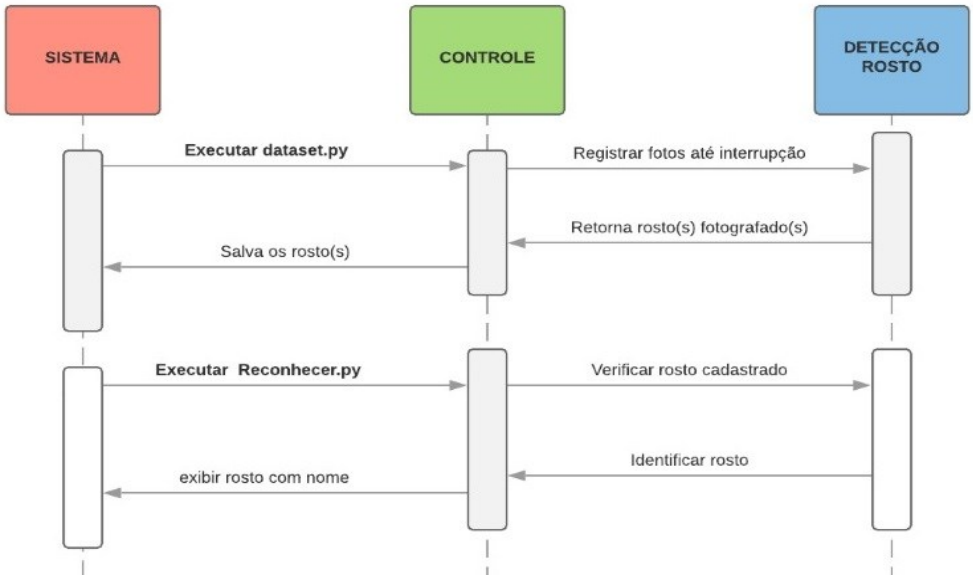

Figura 10 - Diagrama de Sequência

Fonte: Autoria própria (2019).

4.1. REQUISITOS DO SISTEMA

4.1.1. Requisitos Funcionais

Requisitos funcionais que o sistema de reconhecimento facial deve apresentar:

- O sistema deve detectar faces;
- O sistema reconhecerá rostos;
- O sistema de reconhecimento facial deve possuir a funcionalidade de cadastro de pessoas autorizadas;
- As faces cadastradas devem ser apagadas, através de um painel ou manualmente;

4.1.2. Requisitos não funcionais

Requisitos não funcionais que o sistema deve executar:

- O sistema de reconhecimento facial ou o sistema operacional deve possuir autenticação, evitando o acesso indevido;
- O sistema precisa reconhecer somente pessoas autorizadas
- As pessoas cadastradas devem ser inseridas somente por pessoas autorizadas
- O sistema deve identificar as pessoas que não estão cadastradas como "desconhecido"

4.2. Especificação Caso De Uso

A figura 11, mostra os atores do uso de caso UML, onde o administrador faz os cadastros e o segurança pode visualizar os funcionários ou pessoas cadastradas no sistema.

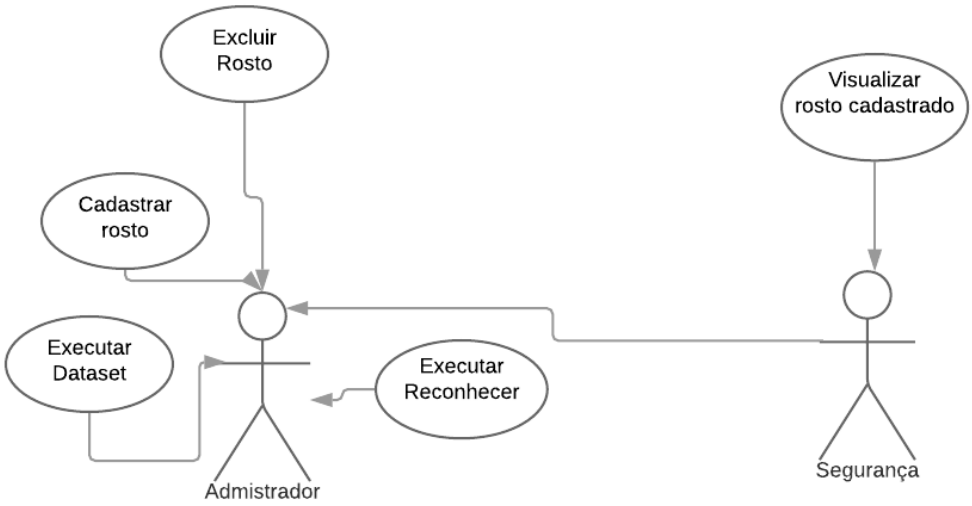

Figura 11 - Uso de caso UML

Fonte: Autoria própria (2019).

Ator: Administrador

Objetivo: Executar Dataset

Pré-condições

O administrador deve estar logado no sistema operacional e o terminal aberto.

1- O administrador executa o comando dataset

Ator: Administrador

Objetivo: Executar Reconhecer

Pré-condições

O administrador deve estar logado no sistema operacional e o terminal aberto.

1- O administrador executa o programa reconhecer.py

Ator: Administrador

Objetivo: Cadastrar rosto

Pré-condições

O administrador deve estar logado no sistema operacional e o terminal aberto.

1- Executar o programa dataset.py
2- Aguardar quantidade mínimo de 10 faces
3- Renomear a face identificada
4- Salva rosto conhecido

Ator: Administrador

Objetivo: excluir rosto

Pré-condições

O administrador deve estar logado com usuário como administrador no sistema operacional e o terminal aberto.

1- Entrar na pasta raiz do programa.
2- Entrar na pasta pessoas.
3- Deletar o arquivo com nome ou identificação da pessoa.

Ator: Segurança

Objetivo: Visualizar rosto cadastrado

Pré-condições

O administrador deve estar logado com usuário como administrador no sistema operacional e o terminal aberto.

1- Executar Reconhecer

4.3. Fundamentos da metodologia

4.3.1. Legislação

A Lei geral de proteção de dados exige que se os dados de alguém for armazenado ela precisa ter o consentimento. O sistema de reconhecimento facial proposto deve registrar apenas a presença de quem está autorizado a gravar dados, sendo assim tornando uma ferramenta que desempenha um papel diferente das câmeras de vigilância, atendendo a legislação vigente, lei nº 13.709 de agosto de 2018.

4.3.2. Segurança Da Informação

Organizações como NIST e ISO exigem alguns padrões de segurança física, estabelecendo a necessidade do uso de vigilantes, mas estes vigilantes possuem falhas humanas. Para auxiliar um vigilante que está atendendo padronizações de segurança o sistema de reconhecimento facial é a proposta para maximizar o número de pessoas identificadas em um estabelecimento ou empresa.

4.3.3. Detecção Facial

A detecção facial é um processo importante, pois antes do reconhecimento, identifica a posição do rosto na imagem, este processo facilita remover a maior parte de informação desnecessária para o processo de classificação de imagem. Para o processo de detecção de rosto neste projeto foi identificado a necessidade do Haar Cascade, um algoritmo derivado do Viola Jones com melhorias, um algoritmo de aprendizado de máquina, este algoritmo detecta apenas a frente do rosto, reduzindo o número de detecções que poderiam não ser fáceis de reconhecer, reduzindo o número de imagens que precisam ser processadas.

4.3.4. Reconhecimento Facial

O reconhecimento é um processo pós detecção, pode ser feito utilizando diversas técnicas de análise de ponto de interesse, porém a proposta deste projeto é apenas demonstrar o reconhecimento facial utilizando um algoritmo de classificação, o algoritmo dos vizinhos mais próximos o K-NN sem técnicas processamentos de imagens aprimoradas, este algoritmo utiliza distancia euclidiana, tornando possível dizer o quão próximo é um rosto capturado de um registrado.

4.4. Tecnologias

4.4.1. Python

A linguagem Python é uma linguagem de alto nível, fácil leitura e disponível para múltiplas plataformas, facilitando a portabilidade dos softwares desenvolvidos. Conforme os anexos foi desenvolvido dois programas em Python, um para registrar os rostos das pessoas autorizadas que serão detectadas e outro para ler os rostos registrados encontrados na pasta e comparar uma foto capturada com as conhecidas utilizando o algoritmo dos vizinhos próximos.

4.4.2. OpenCV

A biblioteca do Python de visão computacional (OpenCV) permite manipularmos as imagens e acessar a câmera de forma facilitada através de sua documentação. Após o processo de detecção de face com o algoritmo Haar Cascade, obtém-se as coordenadas cartesianas da localização das faces disponíveis em uma foto capturada, com o uso da biblioteca recorta-se e redimensiona o rosto, ainda utilizando a biblioteca convertemos a foto para um vetor unidimensional, facilitando para salvar e executar o algoritmo de classificação K-NN.

4.5. Recursos De Hardware e Software

4.5.1. Recursos de hardware

4.5.1.1. Notebook

Para a construção, execução e testes do algoritmo de reconhecimento facial proposto foi utilizado um notebook com saída de vídeo e as seguintes especificações:

Processador i7;
8 GB DDR3 de memória RAM;
Placa de vídeo 635M 2GB 128 bits DDR5;
Armazenamento com SSD 256 GB;

4.5.1.2. BeagleBone

Para execução de testes de desempenho em um sistema embarcado foi utilizado a BeagleBone, uma placa de desenvolvimento sem saída de vídeo com o tamanho próximo a de um cartão de crédito e com as seguintes características:

Processador Arm AM335 Cortex-M3 720MHz;
256 MB DDR2 de memória RAM;

4.5.1.3. Webcam

A captura das imagens para o reconhecimento facial foi através de uma webcam Logitech C920 HD PRO com interface usb e resoluções de até 1920 x 1080 pixels.

4.5.2. Recursos de software

4.5.2.1. Ubuntu

O sistema de reconhecimento facial foi desenvolvido e executado em um sistema operacional Ubuntu na versão 18.04 LTS no notebook

4.5.2.2. Debian

Após o sistema de reconhecimento facial ser desenvolvido e executado no Ubuntu na versão 18.04 LTS, também foi testado na placa de desenvolvimento BeagleBone executando um sistema operacional Debian embarcado.

4.6. Viabilidade

O projeto de reconhecimento facial é de baixo custo pois pode ser executado em placas com sistemas operacionais embarcados. O Código desenvolvido pode ser executado em Python na versão 2.7 ou superior, disponível em diversos sistemas. A placa utilizada BeagleBone infelizmente não está mais disponível para aquisição nas lojas, mas uma versão superior conhecida como BeagleBone Black está sendo vendida por aproximadamente R$250 no aliexpress.com (loja chinesa).

O projeto foi testado com uma webcam logitech que é vendida por aproximadamente por R$250 na internet por lojas brasileiras, mas versões que suportam o 720p podem ser encontradas por R$89.

5. CONTEXTO

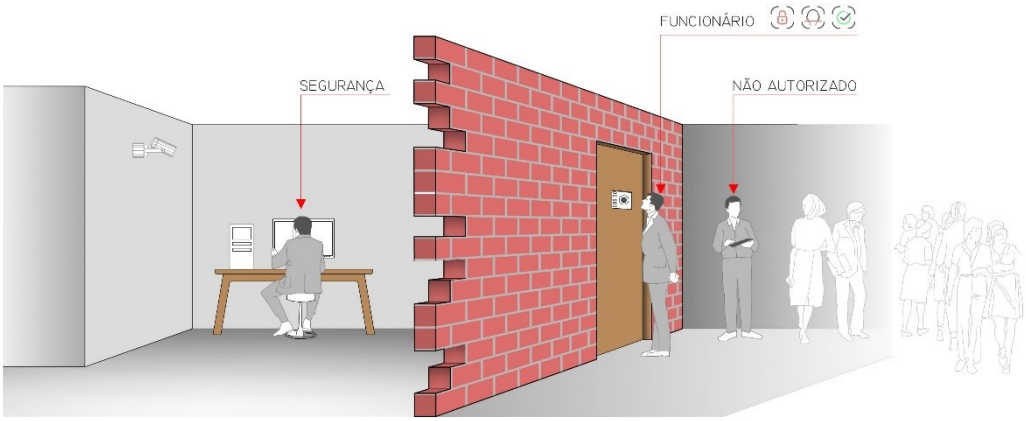

Figura 12 - Ilustração de cenário

Fonte: Autoria própria (2019).

Um caso de uso para o sistema de reconhecimento facial é em empresas que possuem acessos restritos, normalmente empresas que são rígidas quanto a segurança física estão atendendo legislações e padronizações de segurança como NIST (não regulatória), PCI (regulatória) e ISO 27000 (regulatória). A figura 12 ilustra uma requisição de autorização de um funcionário através da câmera na porta, ao ser identificado e autorizado, o sistema libera a porta, mas pode acontecer de uma pessoa não autorizada aguardar a porta ser liberada para realizar o *Tailgating*, técnica utilizada por invasores para aproveitar a liberação do acesso, como demonstrado na figura 12 o espaço a ser acessado é composto por uma câmera, validando e exibindo o rosto dos funcionários para o vigilante no computador, sendo assim se caso houver uma invasão a pessoa não autorizada ainda poderá ser contido pelo segurança.

6. RESULTADOS

Para realizar o reconhecimento é necessário ter pelo menos um candidato a comparação registrado, caso contrário o sistema de reconhecimento irá fechar, após o cadastro de pelo menos um autorizado, é possível executar o código de reconhecimento e caso o rosto seja desconhecido será marcado em cor vermelha como demonstrado na figura 13 registrada no Ubuntu versão 18.04 LTS com interface gráfica.

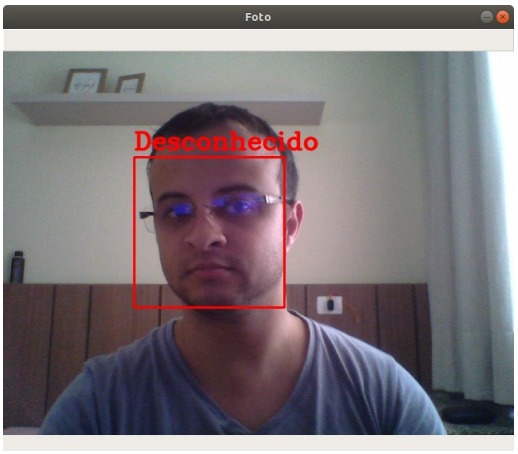

Figura 13 - Sistema detectando, mas não reconhecendo

Fonte: Autoria própria (2019).

Caso a face detectada seja registrada ela será demarcada e irá apresentar o nome da pessoa, como demonstrado na figura 14.

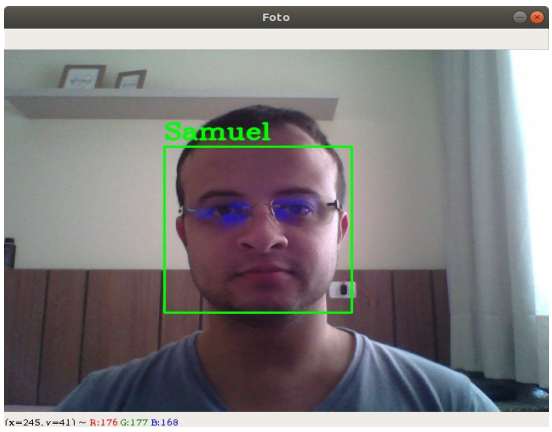

Figura 14 - Sistema detectando, mas reconhecendo

Fonte: Autoria própria (2019).

Nome	Período	Total	Acertos	Erros	Precisão
Andrey	Manhã	150	82	68	54,67%
Andrey	Tarde	150	103	47	68,67%
Andrey	Noite	150	77	73	51,33%
Samuel	Manhã	150	79	71	52,67%
Samuel	Tarde	150	98	52	65,33%
Samuel	Noite	150	76	74	50,67%
				Média	57,22%

Tabela 3 - Precisão dos Resultados

Fonte: Autoria própria (2019).

Conforme a tabela 3, a primeira coluna “Nome” é dos participantes para o teste, na segunda coluna “Período” foram feitos os testes durante os períodos manhã, tarde e noite, cada período possui diferente intensidade de luz. Na terceira coluna “Total” foi a quantidade de fotos que foi tirada de cada pessoa para teste, a coluna quatro “Acertos” é a quantidade de fotos que foram identificadas corretamente o nome da pessoa sendo testada, na quinta “Erros” quantas vezes o algoritmo não identificou a pessoa ou identificou como outra pessoa. Com base nos erros e acerto foi feito as precisões conforme o cálculo abaixo.

$$Precisão = \frac{Acerto}{Acerto + Erro} \quad (8)$$

6.1. Desempenho

A placa de desenvolvimento com Linux embarcado BeagleBone apresentou uma soma de consumos de *threads* de 56.8% do CPU e memória 42.6% de RAM.

Conforme a figura 15 e 16:

```
debian@beaglebone:~$ ps -o pid,user,%mem,command ax | sort -b -k3 -r
  PID USER     %MEM COMMAND
 1510 debian   42.6 python reconhecer.py
```

Figura 15 - Demonstra o consumo memória

Fonte: Autoria própria (2019).

```
 360M 99936 52208 S  0.0 42.5  0:00.00 python reconhecer.py
 360M 99936 52208 R 11.9 42.5  0:00.61 python reconhecer.py
 360M 99936 52208 R 11.9 42.5  0:00.64 python reconhecer.py
 360M 99936 52208 R 10.6 42.5  0:00.57 python reconhecer.py
 360M 99936 52208 S 11.2 42.5  0:00.53 python reconhecer.py
 360M 99936 52208 R 11.2 42.5  0:00.54 python reconhecer.py
 360M 99936 52208 S 11.2 42.5  0:00.56 python reconhecer.py
 360M 99936 52208 R 11.2 42.5  0:00.57 python reconhecer.py
 360M 99936 52208 S 10.6 42.5  0:00.53 python reconhecer.py
```

Figura 16 - Demonstra 5 threads ativas

Fonte: Autoria própria (2019).

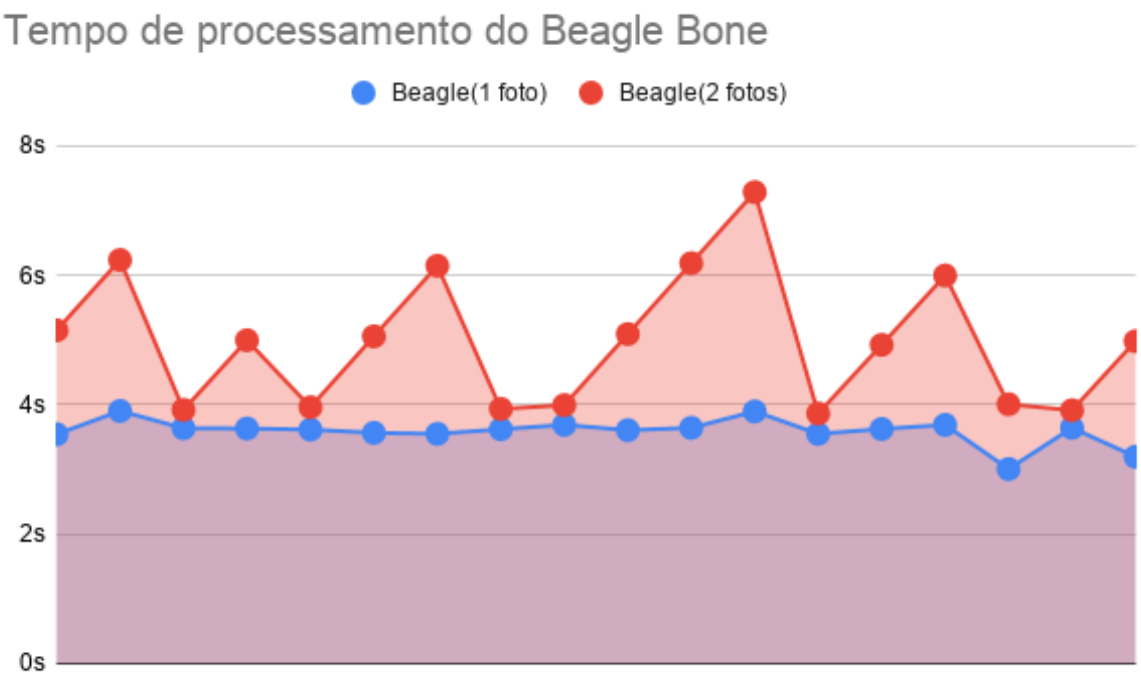

Figura 17 - Tempo de processamento Beaglebone

Fonte: Autoria própria (2019).

O Notebook com Ubuntu apresentou uma soma de consumos de *threads* de 73.9% do CPU e memória 1.6% de RAM.

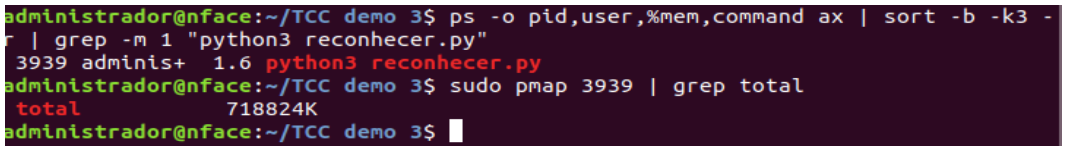

Figura 18 - Demonstra o consumo memória

Fonte: Autoria própria (2019).

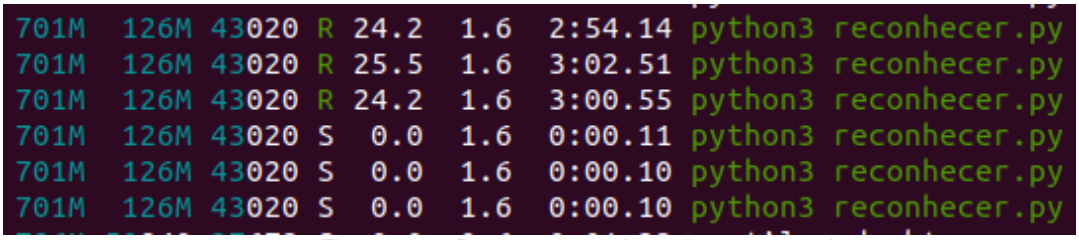

Figura 19 - Demonstra 3 threads executando

Fonte: Autoria própria (2019).

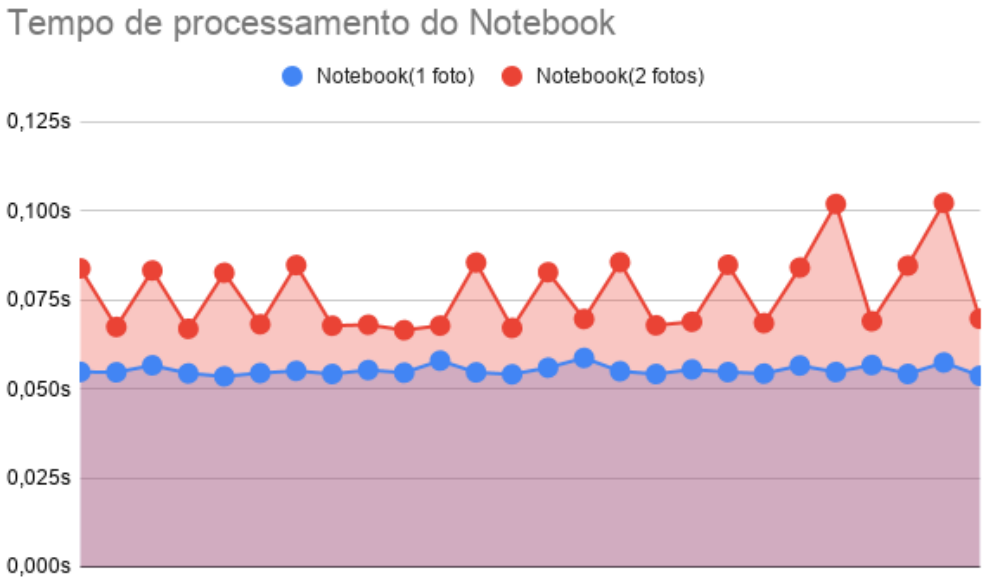

Figura 20 - Tempo de processamento Notebook

Fonte: Autoria própria (2019).

6.2. Análise De Riscos

A falha humana ainda não irá deixar de existir com o uso do sistema, pois a proposta de reconhecimento facial é para auxiliar o vigilante, outro problema é que qualquer sistema de reconhecimento facial não é totalmente preciso pois ele trabalha através de cálculos de proximidades das faces detectadas. Durante o desenvolvimento do projeto também se identificou que o sistema pode ser enganado através da exibição de uma fotografia, mas para solucionar este

problema envolve estudos específicos, exigindo a pesquisa de trabalhos futuros para tratar o problema de falsificação de identidade. O projeto visa resolver alguns problemas regulatórios, mas com o desenvolvimento foi identificado a necessidade de proteção dos dados pessoais localmente, porém não foi implementado qualquer mecanismo de proteção de dados gerando novos problemas que precisam ser tratados em caso de uso real.

7. CONCLUSÕES

O sistema proposto não possui uma precisão alta, porém exige baixo processamento, o desenvolvimento demonstrou a possibilidade do uso e o processamento exigido. Este projeto atende as regulamentações parcialmente podendo substituir um sistema de CFTV em caso de circulação de pessoas que não permitem o registro de imagem, atendendo a lei geral de proteção de dados.

8. REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL. LEI Nº 605, DE 5 DE JANEIRO DE 1949. **Repouso semanal remunerado e o pagamento de salário nos dias feriados civis e religiosos.**, Jan 1949. Disponivel em: <http://www.planalto.gov.br/ccivil_03/leis/L0605.htm>. Acesso em: 10 out. 2019.

BRASIL. LEI Nº 13.709, DE 14 DE AGOSTO DE 2018. **Lei Geral de Proteção dos Dados**, ago 2018. Disponivel em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm>. Acesso em: 10 out. 2019.

CYSEC. **PCI DSS COMPLIANCE**. Disponivel em: <http://pcidsscompliance.net/pci-dss-requirements/how-to-comply-to-requirement-9-of-pci-dss/>. Acesso em: 10 out. 2019.

E., N.; R., Y. **Memória Inteligente**. Petrópolis: Editora Vozes Ltda, 2017.
HINTZBERGEN, J. et al. **Fundamentos de Segurança da Informação**. 3. ed. [S.l.]: BRASPORT, 2015.

ISMS.ONLINE. Disponivel em: <https://www.isms.online/iso-27001/annex-a-11-physical-and-environmental-security/>. Acesso em: 10 out. 2019.

TECHPEDIA. **Facial Recognition**. Disponível em: < https://www.techopedia.com/definition/32071/facial-recognition > Acesso em: 30 de setembro de 2019.

KLEINA. **Como funcionam os sistemas de reconhecimento facial**. Disponível em: < https://www.tecmundo.com.br/camera-digital/10347-como-funcionam-os-sistemas-de-reconhecimento-facial.htm> Acesso em: 30 de setembro de 2019.

SILVA J.I.S. **RECONHECIMENTO FACIAL EM IMAGENS DE BAIXA RESOLUÇÃO.** Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/16367/1/disserta%C3%A7%C3%A3o_jiss_ci%C3%AAnciadacomputa%C3%A7%C3%A3o.pdf> Acesso em: 03 de novembro de 2019.

BISSON. D **5 ataques de engenharia social a serem observados.** Disponível em: < https://www.tripwire.com/state-of-security/security-awareness/5-social-engineering-attacks-to-watch-out-for/ > Acesso em: 03 de novembro de 2019.

VERKADA. **Payments Security and Video Surveillance: What to Know.** Disponível em: <https://www.verkada.com/blog/payments-security-video-surveillance/> Acesso em: 03 de novembro de 2019.

DIGITALGUARDIAN. **DEFINITION OF NIST SP 800-53.** Disponível em: <https://digitalguardian.com/blog/what-nist-sp-800-53-definition-and-tips-nist-sp-800-53-compliance> Acesso em: 03 de novembro de 2019.

NIST. **About NIST.** Disponível em: <https://www.nist.gov/about-nist> Acesso em: 03 de novembro de 2019.

PROOF. **Como funciona a LGPD.** Disponível em: <https://www.proof.com.br/blog/como-funciona-a-lgpd/> Acesso em: 03 de novembro de 2019.

DINO. **Reconhecimento facial revoluciona controle de acesso e segurança predial.** Disponível em: <https://www.terra.com.br/noticias/dino/reconhecimento-facial-revoluciona-controle-de-acesso-e-seguranca-predial,44331b2b7607f2a9733e3667420f30besm3h4quj.html> Acesso em: 03 de novembro de 2019.

CARVALHO, A. P. L. F. **Redes Neurais Artificiais.** Disponível em: <http://conteudo.icmc.usp.br/pessoas/andre/research/neural/> Acesso em: 03 de novembro de 2019.

TOSCANO.R, **Integral Image** Disponível em: <https://tecnoblog.net/247956/referencia-site-abnt-artigos/>. 01 dezembro de 2019.

CLÉSIO.F, **ENCICLOPÉDIA DAS DISTÂNCIAS (MICHEL DEZA & ELENA DEZA)** Disponível em: < https://mineracaodedados.wordpress.com/tag/distancia-euclidiana/>. 01 dezembro de 2019.

COSTA, T. F. **RECONHECIMENTO FACIAL COM HAAR CASCADE E K-NN**. UNIVERSIDADE FEDERAL RURAL DO SEMI-ARIDO CAMPUS PAU DOS FERROS. Mossoró, Rio Grande do Norte, p. 53. 2018.

R., Fernando. J. V. Z. &. R. Projeto de Redes Neurais Artificiais. Unicamp. [S.l.]. 2019.

ANEXO

gerarDataset.py

```
import cv2
import numpy as np
#Abre a câmera
camera = cv2.VideoCapture(0)
#Inicializa o detector de face
#Arquivo disponivel na biblioteca OpenCV
detRostos = cv2.CascadeClassifier("treinamento_rostos.xml")
#Armazena os rostos em um array temporario
rostos = []
#Pasta das faces
pasta = './pessoas/'
#Nome do arquivo
pessoa = input("Nome: ")
flag2save = 0
while True:
  try:
        #Captura o frame
        fotoRecebida,foto = camera.read()

        #fotoRecebida é uma variável de controle para verificar se o frame foi recebido
        if fotoRecebida==False:
                print("Foto não recebida")
                continue
        #Rostos detectados Ex: (x,y,largura,altura)
        rostosDetectados = detRostos.detectMultiScale(foto,1.3,5)
        if len(rostosDetectados)==0:
                continue
        #Organiza as posições dos rostos em ordem crescente de tamanho
        rostosDetectados = sorted(rostosDetectados,key=lambda f:f[2]*f[3])
        #Marca e recorta o último rosto detectado
        for rosto in rostosDetectados[-1:]:
                x,y,w,h = rosto
                cv2.rectangle(foto,(x,y),(x+w,y+h),(0,0,255),1)
                #O offset é a quantidade de pixel lateral que sera adicionado a posição do rosto
identificada
                offset = 10
                #recorta o rosto
                rosto_recortado = foto[y-offset:y+h+offset,x-offset:x+w+offset]
                #Normaliza para resolução de 100x100
                rosto_recortado = cv2.resize(rosto_recortado,(100,100))
                flag2save += 1
                #Salva o rosto a cada 5 fotos
                if flag2save%5==0:
                        rostos.append(rosto_recortado)
                        if (len(rostos) == 1):
                                print(str(len(rostos)) + " rosto salvo")
```

```
                        if(len(rostos) > 1):
                                print(str(len(rostos)) + " rostos salvos")
            #Mostra somente o rosto
            cv2.imshow("Foto",foto)
            #Mostra a foto
            cv2.imshow("Rosto",rosto_recortado)
            #Espera tecla
            esperaTecla = cv2.waitKey(1)
    except KeyboardInterrupt:
            break
#Salva a list do tipo tuple em um array numpy
rostos = np.asarray(rostos)
#Remove valor da matriz com tipo de valores
rostos = rostos.reshape((rostos.shape[0],-1))
#Salva todos os rostos capturados em um arquivo de uma pessoa
np.save(pasta+pessoa+'.npy',rostos)
#Libera a camera
camera.release()
#Fecha as janelas
cv2.destroyAllWindows()
```

reconhecer.py

```
import numpy as np
import os, time, cv2
def distancia(f1, f2):
  return np.sqrt(((f1-f2)**2).sum())
def knn(foto_treino, foto_detectado, k=5):
  distancias = []
  for i in range(foto_treino.shape[0]):
          ix = foto_treino[i, :-1]  #Foto
          iy = foto_treino[i, -1]   #Identificador unico
          #Calcula a distancia
          d = distancia(foto_detectado, ix)
          distancias.append([d, iy])
  #Organiza em ordem crescente
  distk = sorted(distancias, key=lambda x: x[0])[:k]
  #Pega o ponto mais distante dos 5
  proximidade = np.amax(distk, axis=0)
  if(proximidade[0] > 9000):
          return -1
  #Identificadores disponíveis
  labels = np.array(distk)[:, -1]
  #identificadores Unicos
  idUnico = np.unique(labels, return_counts=True)
  #Quantidade de correspondencia
  index = np.argmax(idUnico[1])
  return idUnico[0][index]
#Abre a camera
camera = cv2.VideoCapture(0)
#detecta rosto
#Arquivo disponivel na biblioteca OpenCV
detRostos = cv2.CascadeClassifier("treinamento_rostos.xml")
pasta = './pessoas/'
# Faces detectadas
rostos = []
# identificadores
labels = []
_id = 0 # id para cada nome
nomes = {}  #Array para mapear nomes com os id
#Lista e le todos os arquivos de rostos disponíveis
for arquivo in os.listdir(pasta):
  if arquivo.endswith('.npy'):
          #Cria um mapa de nome por ID
          nomes[_id] = arquivo[:-4] #Retorna nome do arquivo sem .npy
          data_item = np.load(pasta+arquivo)
          rostos.append(data_item)
          #Cada ponto in cada arquivo tera a label incrementada
          target = _id*np.ones((data_item.shape[0],))
          _id += 1
          labels.append(target)
```

```
	#Concatena os valores dos treinamento
	face_dataset = np.concatenate(rostos,axis=0)
	face_labels = np.concatenate(labels,axis=0).reshape((-1,1))
	#Concatena o dataset com os identificadores unicos
	trainset = np.concatenate((face_dataset,face_labels),axis=1)
	while True:
	  #Captura o frame
	  inicioT = time.time()
	  fotoRecebida,foto = camera.read()
	  #fotoRecebida é uma variável de controle para verificar se o frame foi recebido
	  if fotoRecebida==False:
			print("Foto não recebida")
			continue
	  #Rostos detectados Ex: (x,y,largura,altura)
	  rostosDetectados = detRostos.detectMultiScale(foto,1.3,5)
	  if(len(rostosDetectados)==0):
		continue
	  for rosto in rostosDetectados:
		x,y,w,h = rosto
		#O offset é a quantidade de pixel lateral que sera adicionado a posição do rosto
identificada
		offset = 10
		rosto_recortado = foto[y-offset:y+h+offset,x-offset:x+w+offset]
		#Normaliza para resolução de 100x100
		rosto_recortado = cv2.resize(rosto_recortado,(100,100))
		#Reconhece
		out = knn(trainset,rosto_recortado.flatten())
		if(out == -1):
			nome = "Desconhecido"
			cv2.putText(foto,nome,(x,y-
10),cv2.FONT_HERSHEY_COMPLEX,1,(0,0,255),2,cv2.LINE_AA)
			cv2.rectangle(foto,(x,y),(x+w,y+h),(0,0,255),2)
			fimT = time.time()
			segundos = fimT - inicioT
			print(int(1/segundos))
		else:
			fimT = time.time()
			segundos = fimT - inicioT
			print(int(1/segundos))
			nome = nomes[int(out)]
			cv2.putText(foto,nome,(x,y-
10),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2,cv2.LINE_AA)
			cv2.rectangle(foto,(x,y),(x+w,y+h),(0,255,0),2)
	  #cv2.imshow("Foto",foto)
	  #Espera tecla
	  esperaTecla = cv2.waitKey(1)
	#Libera a camera
	camera.release()
	#Fecha as janelas
	cv2.destroyAllWindows()
```