

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Francisco Tassinari Fajardo

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB COM DETECÇÃO E  
RECONHECIMENTO FACIAL**

Santa Maria, RS  
2019

**Francisco Tassinari Fajardo**

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB COM DETECÇÃO E  
RECONHECIMENTO FACIAL**

Tese de Bacharelado apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

ORIENTADOR: Prof. João Vicente Ferreira Lima

---

©2019

Todos os direitos autorais reservados a Francisco Tassinari Fajardo. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

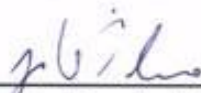
End. Eletr.: [fffajardo@gmail.com](mailto:fffajardo@gmail.com)

Francisco Tassinari Fajardo

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB COM DETECÇÃO E  
RECONHECIMENTO FACIAL**

Tese de Bacharelado apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

**Aprovado em 5 de dezembro de 2019:**

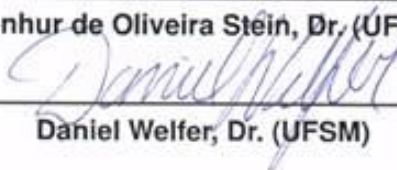


\_\_\_\_\_  
**João Vicente Ferreira Lima, Dr. (UFSM)**

(Presidente/Orientador)



\_\_\_\_\_  
**Benhur de Oliveira Stein, Dr. (UFSM)**



\_\_\_\_\_  
**Daniel Welfer, Dr. (UFSM)**



## RESUMO

# DESENVOLVIMENTO DE UMA APLICAÇÃO WEB COM DETECÇÃO E RECONHECIMENTO FACIAL

AUTOR: Francisco Tassinari Fajardo

ORIENTADOR: João Vicente Ferreira Lima

Os seres humanos realizam de maneira natural o ato de detectar e reconhecer faces, mas para um computador a tarefa de descrever objetos a partir de imagens não é algo trivial. A Visão Computacional teve uma revolução após o uso de redes neurais convolucionais para identificação de padrões. Uma área beneficiada é a de segurança visto que os sistemas que exploram a detecção de faces e o reconhecimento das mesmas tiveram um aumento de precisão. Este trabalho desenvolve uma aplicação web para a visualização e organização de informações sobre métodos conhecidos na literatura, métodos de detecção e de reconhecimento facial. Além disso, o trabalho usa imagens retiradas de câmeras de segurança, e demonstra as detecções e reconhecimentos de uma maneira organizada na aplicação, uma vez que coletar as informações sobre tais métodos pode ser uma tarefa desgastante. Esse trabalho detalha o sistema implementado que utiliza os seus métodos de bibliotecas públicas que contribuem com o avanço da área e comenta os resultados obtidos. A literatura nessa área geralmente trabalha com ambientes controlados e imagens de alta resolução para obter bons resultados, esse trabalho descreve seus resultados a partir de testes realizados em conjuntos de imagens já existentes e também imagens coletadas da câmera de segurança. Os testes de detecção nos conjuntos de imagens prontas avaliam a capacidade dos diferentes métodos de detectar mudanças de ângulo e iluminação, e conclui-se que ângulos extremos não são detectados usando o método de rede neural convolucional e a iluminação é um fator determinante na detecção. Nos conjuntos de imagens da câmera a detecção é avaliada sobre os diferentes métodos, porém sem variações extremas de ângulo ou iluminação, sendo que nas distâncias estipuladas obtém-se 100% de detecção. Os testes de reconhecimento em um conjunto de imagens prontas exibem a dificuldade do reconhecimento em situações reais, nas imagens da câmera foi possível avaliar as distâncias em que o reconhecimento começa a conter falsos-negativos para uma mesma pessoa devido à baixa resolução de uma das fotos.

**Palavras-chave:** Visão Computacional. Web. Detecção Facial. Reconhecimento Facial. Redes Neurais Convolucionais.

## ABSTRACT

### DEVELOPMENT OF A WEB APPLICATION WITH FACIAL RECOGNITION AND DETECTION

AUTHOR: Francisco Tassinari Fajardo

ADVISOR: João Vicente Ferreira Lima

Humans naturally perform the act of detecting and recognizing faces, but for a computer the task of describing objects from images is not trivial. Computer Vision had a revolution after the use of convolutional neural networks to identify patterns. A field benefited is the security field since the systems that exploit the detection of faces and the recognition of them had an increase of precision. This work develops a web application for viewing and organizing information about methods known in the literature, methods of detection and of facial recognition. In addition, the work uses images taken from security cameras, and demonstrates detections and recognitions in an organized manner in the application, since gathering information about such methods can be a exhausting task. This work details the implemented system that uses its methods of public libraries that contribute to the advance of the field and comments the obtained results. Literature in this area generally works with controlled environments and high resolution images to obtain good results, this work describes its results from tests performed on existing image sets and also images collected from the security camera. The detection tests on ready image sets assess the ability of different methods to detect changes of angle and illumination, and it is concluded that extreme angles are not detected using the convolutional neural net method and illumination is a determining factor in detection. In camera image sets detection is evaluated on different methods, but without extreme angle or illumination variations, and at the stipulated distances 100% detection is obtained. The recognition tests on a set of ready images show the difficulty of recognition in real situations, in camera images it was possible to evaluate the distances at which recognition begins to contain false negatives for the same person due to the low resolution of one of the photos.

**Keywords:** Computer Vision. Web. Face Detection. Face Recognition. Convolutional Neural Network

## LISTA DE FIGURAS

Figura 1.1 – Ilustração de algoritmos de detecção de faces e elementos da face. ....	10
Figura 2.1 – Imagem com a face normal e a direita seu vetor HOG que será usado no classificador. ....	15
Figura 2.2 – Exemplificação das etapas do Método HOG após transformação na imagem de entrada. ....	16
Figura 2.3 – Desempenho do método escolhido no FDBB, o chamado MMOD-HOG em comparação ao HOG tradicional. ....	16
Figura 2.4 – Representação clássica do modelo Perceptron de rede neurais. ....	17
Figura 2.5 – Representação de uma arquitetura de redes neurais convolucionais. ...	19
Figura 2.6 – Representação de funcionamento do MTCNN e dos resultados entre sub-redes. ....	20
Figura 2.7 – Arquitetura do MTCNN. ....	21
Figura 2.8 – Gráfico superior esquerdo representa o FDBB e os outros gráficos os diferentes sets do WIDER. ....	21
Figura 2.9 – Descrição do processo de Reconhecimento facial. ....	24
Figura 2.10 – Erro de treinamento (esquerda) e erro de teste (direita) em redes com 20 e 56 camadas. ....	25
Figura 3.1 – Diagrama de casos de uso representando as ações de detectar e reconhecer faces. ....	30
Figura 3.2 – Interface inicial da aplicação. ....	30
Figura 3.3 – Representação da tela de detecção com um erro de validação. ....	32
Figura 3.4 – Tela de gerência de configurações para detecção. ....	32
Figura 3.5 – Tela de dados individuais de detecção. ....	33
Figura 3.6 – Exemplo de visualização das fotos e suas detecções. ....	34
Figura 3.7 – Diagrama de atividades para detecção. ....	35
Figura 3.8 – Tela de reconhecimento com botões. ....	37
Figura 3.9 – Tela de Desconhecidos (faces no banco). ....	37
Figura 3.10 – Tela de distâncias de tentativas de reconhecimento. ....	38
Figura 3.11 – Diagrama de atividades para reconhecimento. ....	39
Figura 3.12 – Representação do diretório de um dia e das faces armazenadas pelo sistema no dia. ....	40
Figura 4.1 – Exemplo de imagem do <i>dataset</i> utilizado com variação de iluminação e pose. ....	44
Figura 4.2 – Representação do <i>dataset</i> de medidas circulares em diferentes distâncias, no qual a primeira marcação visível a esquerda é a de 3 metros com uma inclinação de 0° a câmera, e o indivíduo está a 11° para direita. ...	47
Figura 4.3 – Representação dos pontos de medidas do <i>dataset</i> de altas distâncias, indivíduo na posição mais distante em 8,3 metros. ....	47
Figura 5.1 – Exemplo de imagem da interface numa comparação de expressão facial. ....	52
Figura A.1 – Exemplo de fotos do <i>dataset</i> amostrado circularmente em 2 metros de distância da câmera. ....	57
Figura A.2 – Exemplo de fotos do <i>dataset</i> de distâncias maiores com exemplo de 6 e 7 metros de distância para a câmera. ....	58

## LISTA DE GRÁFICOS

Gráfico 2.1 – Comparação de métodos de detecção no FDDB, eixo vertical representa a taxa de faces encontradas que realmente existiam e eixo o horizontal a quantidade de falsas faces que foram encontradas. .... 14

## LISTA DE QUADROS

Quadro 4.1 – Métricas obtidas no banco de imagens com variação de ângulo na face de uma pessoa. ....	42
Quadro 4.2 – Métricas obtidas no banco de imagens com variação de iluminação e ângulo. ....	43
Quadro 4.3 – Métricas obtidas do reconhecimento com tamanho alterado para 50x50 <i>pixels</i> no <i>dataset</i> de reconhecimento e variação da distância euclidiana máxima para reconhecimento. ....	43
Quadro 4.4 – <i>Dataset</i> para distâncias maiores que seis metros, sendo a distância real considerada, o número de imagens e os pontos de amostragem. ....	45
Quadro 4.5 – <i>Dataset</i> com amostragem circular e com mudança do ângulo entre posição da câmera e pessoa, com crescimento do ângulo sentido horário. ..	46
Quadro 4.6 – Tamanho médio das imagens a partir da detecção por distância no <i>dataset</i> amostrado circularmente. ....	46
Quadro 4.7 – Métricas obtidas do HOG com <i>sliding window</i> de 40x40 <i>pixels</i> no <i>dataset</i> amostrado circularmente. ....	48
Quadro 4.8 – Métricas obtidas do MTCNN com <i>sliding window</i> de 40x40 <i>pixels</i> no <i>dataset</i> amostrado circularmente. ....	48
Quadro 4.9 – Métricas obtidas do MTCNN com <i>sliding window</i> 20x20 <i>pixels</i> no <i>dataset</i> amostrado circularmente. ....	48
Quadro 4.10 – Métricas obtidas do MTCNN com <i>sliding window</i> 20x20 <i>pixels</i> no <i>dataset</i> de distâncias maiores que 6 metros e 6 imagens por distância. ....	49
Quadro 4.11 – Métricas obtidas do reconhecimento do <i>set</i> de amostragem circular, com diferentes tamanhos da imagem usando interpolação bicúbica e com distância euclidiana menor ou igual que 0,6. ....	50
Quadro 4.12 – Métricas obtidas do reconhecimento nos conjuntos que foram amostrados circularmente, com variação de tamanho da imagem usando interpolação bicúbica e distância euclidiana menor ou igual que 0,6. ....	50

## LISTA DE ABREVIATURAS E SIGLAS

<i>MTCNN</i>	<i>Multitask Convolutional Neural Network</i> (Rede Neural Convolutacional de Múltiplas Tarefas)
<i>ROI</i>	<i>Region of Interest</i> (Região de Interesse)
<i>FDDB</i>	<i>Face Detection Data Set and Benchmark</i> (Conjunto de Dados e Testes de Desempenho para Detecção de Faces)
<i>ROC</i>	<i>Receiver Operating Characteristic</i> (Característica de Operação do Receptor)
<i>LBPH</i>	<i>Local Binary Pattern Histograms</i> (Padrão Binário Local com Histogramas)
<i>LFW</i>	<i>Labeled Faces in the Wild</i> (Rostos Marcados em Campo)
<i>URL</i>	<i>Uniform Resource Locator</i> (Localizador de recursos uniforme)
<i>UML</i>	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
<i>HOG</i>	<i>Histogram of Oriented Gradients</i> (Histograma de Gradientes Orientados)

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	OBJETIVO	11
<b>1.1.1</b>	<b>Objetivos Específicos</b>	<b>11</b>
1.2	JUSTIFICATIVA	11
1.3	METODOLOGIA	11
1.4	ORGANIZAÇÃO DO TEXTO	12
<b>2</b>	<b>MÉTODOS DE DETECÇÃO E RECONHECIMENTO FACIAL</b>	<b>13</b>
2.1	MÉTODOS DE DETECÇÃO	13
<b>2.1.1</b>	<b>HOG</b>	<b>14</b>
<b>2.1.2</b>	<b>Redes Neurais</b>	<b>17</b>
<b>2.1.3</b>	<b>Redes Neurais Convolucionais</b>	<b>18</b>
<b>2.1.4</b>	<b>MTCNN</b>	<b>19</b>
<b>2.1.5</b>	<b>Métricas de Classificadores Binários</b>	<b>21</b>
2.2	MÉTODOS DE RECONHECIMENTO	23
<b>2.2.1</b>	<b>Reconhecimento na Dlib</b>	<b>24</b>
<b>2.2.2</b>	<b>Fatores Limitantes</b>	<b>25</b>
<b>2.2.3</b>	<b>Trabalhos Relacionados</b>	<b>26</b>
<b>3</b>	<b>SISTEMA DE DETECÇÃO E RECONHECIMENTO DESENVOLVIDO</b>	<b>29</b>
3.1	CASOS DE USO	29
3.2	DETECÇÃO FACIAL	31
<b>3.2.1</b>	<b>Detalhes Específicos</b>	<b>34</b>
3.3	RECONHECIMENTO FACIAL	36
3.4	ARMAZENAMENTO DE FOTOS E FACES	39
3.5	DETALHES EXTRAS DE IMPLEMENTAÇÃO	40
3.6	BIBLIOTECAS UTILIZADAS	41
<b>4</b>	<b>ANÁLISES</b>	<b>42</b>
4.1	EXPERIMENTOS EM BASES DE IMAGENS PRONTAS	42
<b>4.1.1</b>	<b>Testes de Detecção em bases de imagens prontas</b>	<b>42</b>
<b>4.1.2</b>	<b>Testes de Reconhecimento</b>	<b>43</b>
4.2	<i>DATASET</i> DE UMA CÂMERA DE MONITORAMENTO	44
<b>4.2.1</b>	<b>Testes de detecção</b>	<b>48</b>
<b>4.2.2</b>	<b>Testes de reconhecimento</b>	<b>49</b>
4.3	RESULTADOS OBTIDOS	50
<b>5</b>	<b>CONCLUSÃO</b>	<b>52</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>53</b>
	<b>ANEXO A – EXEMPLOS E LOCAIS DOS SETS</b>	<b>57</b>

## 1 INTRODUÇÃO

A área de Visão Computacional segundo (BALLARD; BROWN, 1982) é a construção de descrições de objetos de uma maneira explícita e significativa a partir de imagens, sendo que essa descrição é importante para reconhecer e manipular os objetos.

O fato é que realizar tal feito não é trivial, pois é necessário entender o funcionamento da visão humana em relação a vários fatores como iluminação e nossa percepção de cores (SZELISKI, 2011), a área teve uma revolução após 2012 quando (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) popularizaram o uso de redes neurais convolucionais para resolução de problemas sobre reconhecimento de imagens com sua rede neural AlexNet.

Desde então diversas técnicas foram propostas que tem como objetivo de aumentar a precisão de acertos sobre detecção de faces (ZHANG et al., 2016), na Figura 1.1 observa-se a aplicação de tal método.

O problema da detecção não é nada simples de resolver, afinal existem vários fatores como ângulo da foto, iluminação, expressão da pessoa, mudança de características físicas, qualidade da foto, que dificultam a exatidão do processo, novos métodos também foram aprimorados para o reconhecimento de faces (SUN et al., 2014).

Figura 1.1 – Ilustração de algoritmos de detecção de faces e elementos da face.



Fonte: Autor.

Recentemente os sistemas desenvolvidos são aplicados em diversas áreas como em dispositivos embarcados ao usar a câmera para tirar uma foto, observa-se a detecção de faces em forma de um marcador retangular, já o reconhecimento de faces é utilizado em vários aplicativos como aplicativos médicos nos quais é possível tirar uma foto de um bebê e verificar se ele tem algum problema genético como síndrome de *down* ou a simples



comparação com imagens em um banco de dados.

Vários sistemas também são usados com robôs e suas interações com humanos, entendendo ou não suas emoções através da face, aplicações mais comuns contam com segurança para identificação de pessoas, e controle de acesso onde o sistema funciona como um identificador biométrico.

## 1.1 OBJETIVO

Desenvolver uma aplicação web para estudo de métodos de detecção e reconhecimento facial com imagens de câmeras de segurança.

### 1.1.1 Objetivos Específicos

- Realizar a análise dos métodos de detecção e reconhecimento facial;
- Descrever a implementação da interface e suas características;
- Aplicar os métodos com imagens de câmeras de segurança e analisar os resultados.

## 1.2 JUSTIFICATIVA

Com a recente revolução na área de Visão Computacional, a partir do uso de redes neurais convolucionais, é importante entender o seu funcionamento e seu respectivo desempenho em aplicações visto que segundo (SZELISKI, 2011) essa área é considerada de todas as tarefas que nós podemos pedir para um computador executar, analisar uma imagem e seus objetos constituintes permanece a mais desafiadora.

Sendo assim parece natural o desenvolvimento de aplicações web que façam uso de tais métodos e possam ser utilizados em ambientes preferencialmente genéricos, o sistema desenvolvido busca ser utilizado no dia a dia para tentar reconhecer se uma pessoa detectada é conhecida ou desconhecida.

## 1.3 METODOLOGIA

Para a implementação do sistema primeiramente foram levantados os requisitos para que a aplicação fosse utilizada no dia a dia, em sequência realizou-se a descrição do

projeto e execução do mesmo.

Na análise dos métodos foram utilizadas métricas da área que servem como base para o desempenho desse tipo de sistema, levando em conta o contexto em que ele foi aplicado e as características dos dados.

Além disso, do ponto de vista da natureza esse trabalho é uma pesquisa exploratória aplicada, e do ponto de vista técnico ele engloba o âmbito de um estudo de caso.

#### 1.4 ORGANIZAÇÃO DO TEXTO

A organização textual do presente trabalho funciona da seguinte maneira:

No Capítulo 2, ocorre a descrição de funcionamento e desempenho de diferentes métodos de detecção de faces, isso é importante pois uma vez que se esse processo tem uma baixa porcentagem de acertos, a aplicação desenvolvida não é eficiente.

No mesmo também ocorre a análise dos métodos de reconhecimento e como realizar o pré-processamento da imagem para obtermos um resultado mais expressivo, além de fatores como a quantidade de amostras necessárias.

No Capítulo 3 ocorre a descrição do sistema de maneira formal com diagramas e detalhes de funcionamento do próprio, as ideias aplicadas e bibliotecas utilizadas.

No Capítulo 4 é apresentado os testes realizados sobre o contexto e seus resultados.

No Capítulo 5 é exposto uma visão geral do trabalho e suas conclusões.

## 2 MÉTODOS DE DETECÇÃO E RECONHECIMENTO FACIAL

Ao longo do tempo na literatura foram propostos diversos métodos de detecção e reconhecimento facial, a necessidade de conhecer e utilizar tais recursos estão presentes no desenvolvimento de aplicações de segurança ou monitoramento, a definição do que é detectar e reconhecer uma face e como funcionam tais recursos são de extrema importância para o desempenho dos sistemas.

### 2.1 MÉTODOS DE DETECÇÃO

O ato de detectar faces é a tarefa de percorrer uma imagem e encontrar as faces contidas nela, os autores do *WIDER FACE benchmark and data set* (YANG et al., 2015) ressaltam que existem quatro diferentes tipos de métodos de detecção de faces, métodos baseados em cascata, métodos baseados em partes, métodos baseados em recursos de canal e métodos baseados em redes neurais.

Modelos baseados em partes são aqueles que localizam características (boca, nariz) de uma imagem e então, vem a posição relativa de tais recursos, e verificam se aquilo é o objeto desejado ou não, já os métodos de recursos do canal analisam imagens nos diferentes canais de cor para descobrir se existe o objeto esperado.

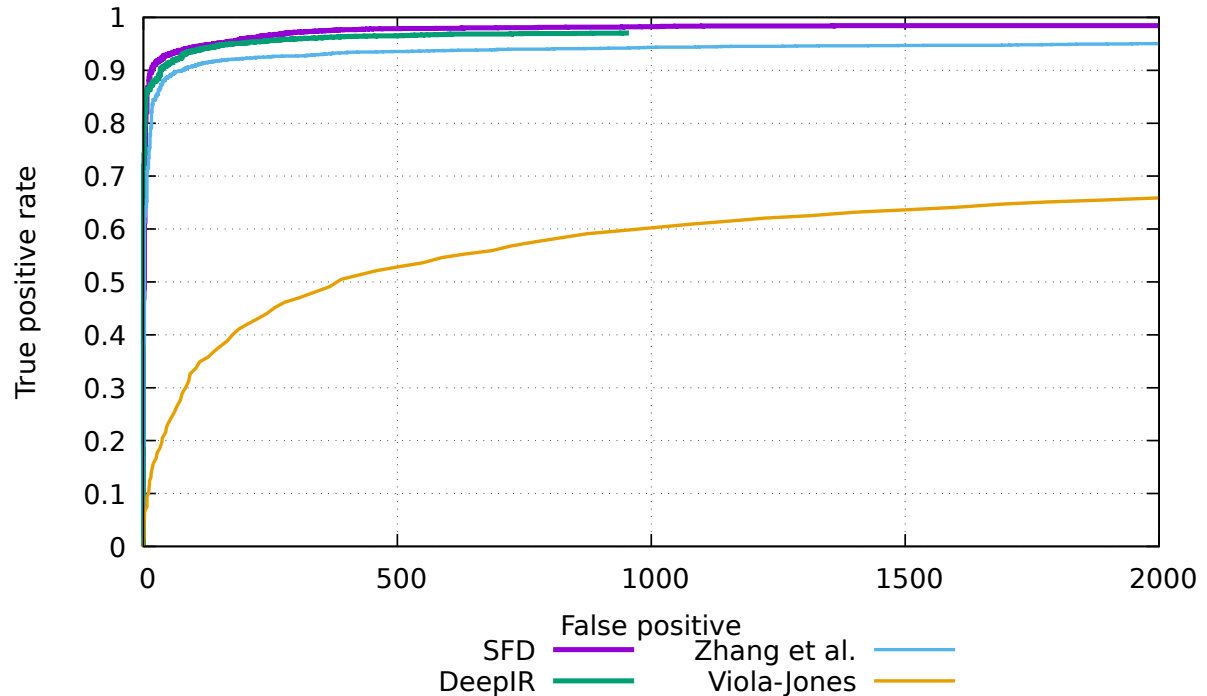
Durante os anos 2000, o método mais conhecido de detecção facial se trata do método de Viola-Jones (VIOLA; JONES, 2001), que pertence ao modelo de cascatas, no qual se extraem características de imagens de pessoas, e de imagens sem pessoas, a fim de analisar os *pixels* e criar um classificador.

Segundo (ZHANG; ZHANG, 2010) a evolução na extração de características de imagens originou métodos de histogramas que ficaram populares, um método discutido que surgiu foi o chamado *Histogram of Oriented Gradients* (HOG), que pode ser aplicado na *interface* e é explicado no Capítulo 2.1.1.

Além do *WIDER FACE* existe o *Face Detection Data Set and Benchmark* (FDDB) (JAIN; LEARNED-MILLER, 2010) que disponibiliza os resultados de testes de detecção de imagem sobre o próprio conjunto de dados, e os métodos que contém o estado da arte atualmente refletem em um grande número de uso de redes neurais.

Observa-se no Gráfico 2.1 a comparação de desempenho (detalhes na Seção 2.1.5) do método já citado Viola-Jones em relação a três métodos que usam redes neurais, sendo que o SFD de (ZHANG et al., 2017) representa um dos melhores métodos atuais, o DeepIR de (SUN; WU; HOI, 2017) apresenta um desempenho próximo e o método utilizado nesse trabalho que é o chamado Rede Neural Convolutacional de Múltiplas Tarefas (MTCNN), explicado no Capítulo 2.1.4.

Gráfico 2.1 – Comparação de métodos de detecção no Fddb, eixo vertical representa a taxa de faces encontradas que realmente existiam e eixo o horizontal a quantidade de falsas faces que foram encontradas.



Fonte: Autor.

### 2.1.1 HOG

O método HOG de (DALAL; TRIGGS, 2005) foi popularizado para detecção de pedestres e vem sofrendo evoluções no seu processo desde sua origem, sendo que sua ideia central consiste de calcular a orientação e magnitude dos gradientes na imagem, colocar esses valores em histogramas para gerar vetores que caracterizam a imagem (Figura 2.1), e então treinar um classificador binário com os valores obtidos, para saber se uma nova imagem contém um objeto específico ou não basta usar o classificador.

Segundo (CERNA, 2014) e (PANCERI et al., 2015) a etapa da criação dos HOGs pode ser dividida em:

1. Normalizar escala e cor, além do cálculo da orientação e magnitude dos gradientes na imagem;
2. Divisão da imagem em blocos e células e cálculo do histograma de orientação dos gradientes por células, sendo os histogramas divididos por  $n$  ângulos de orientação e a inserção no histograma depende da magnitude do gradiente, posteriormente agrupados em blocos;
3. A concatenação dos histogramas que forma o vetor descritor HOG.

Figura 2.1 – Imagem com a face normal e a direita seu vetor HOG que será usado no classificador.



Fonte: Retirado de <<https://www.hackevolve.com/face-recognition-deep-learning/>>.

O gradiente de uma imagem é a variação direcional da intensidade ou cor da mesma, o cálculo da orientação e magnitude dos gradientes na imagem ocorre por meio de uma convolução que usa o chamado filtro de Sobel.

Matematicamente a expressão pode ser descrita como o o gradiente de de x ( $G_x$ ) e y ( $G_y$ ) em função de *pixels* de intensidade  $Y$  nas coordenadas  $x$  e  $y$  da seguinte maneira:

$$G_x(y, x) = Y(y, x + 1) - Y(y, x - 1)$$

$$G_y(y, x) = Y(y + 1, x) - Y(y - 1, x)$$

O passo seguinte é o cálculo da da magnitude  $M_x$  do gradiente e a da orientação  $\theta$  do mesmo, lembrando que a imagem foi repartida por células:

$$M_x(y, x) = \sqrt{G_x(y, x)^2 + G_y(y, x)^2}$$

$$\theta(y, x) = \arctan\left(\frac{G_y(y, x)}{G_x(y, x)}\right)$$

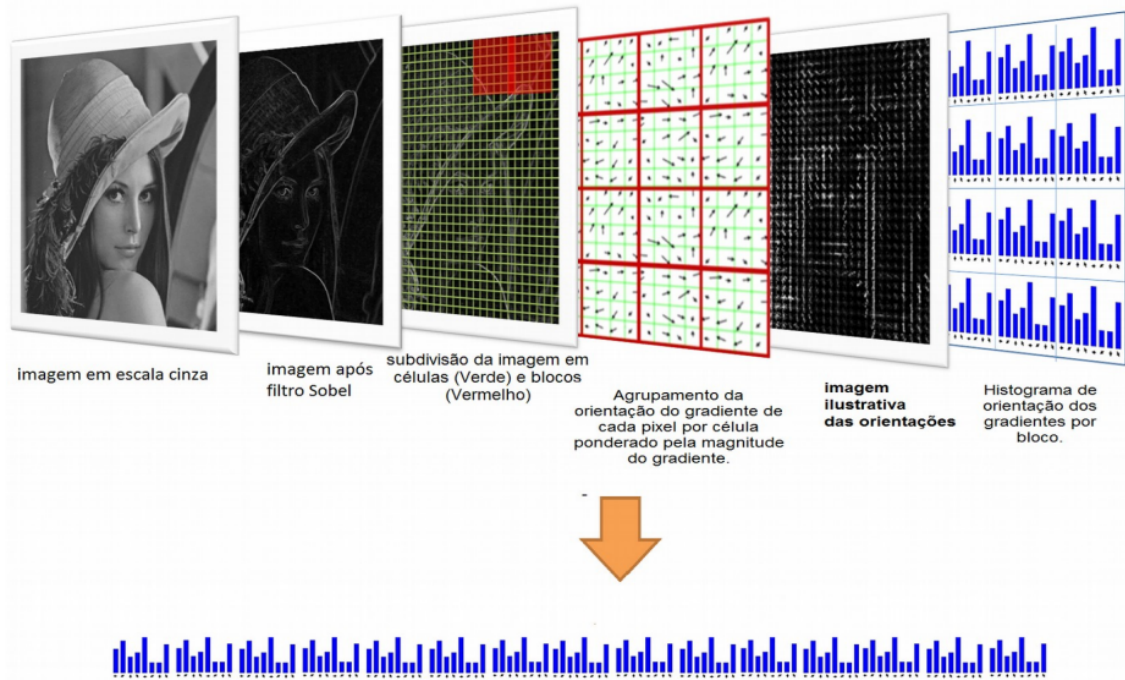
Com a obtenção desses valores ocorre a criação dos histogramas com a intensidade pelo o ângulo, ocorre uma divisão em intervalos de ângulos, usa-se nove intervalos, sendo esse número recomendado pelo autor do método.

O processo de criação do histograma não é tão direto, pois a inserção ocorre com o valor de intensidade sendo dividido de acordo com quanto perto a intensidade está sobre os dois intervalos mais próximos.

Depois de gerado os histogramas ocorre uma normalização de contraste que faz uso dos blocos e gera os valores para serem concatenados e armazenados. O processo completo pode ser visualizado na Figura 2.2.

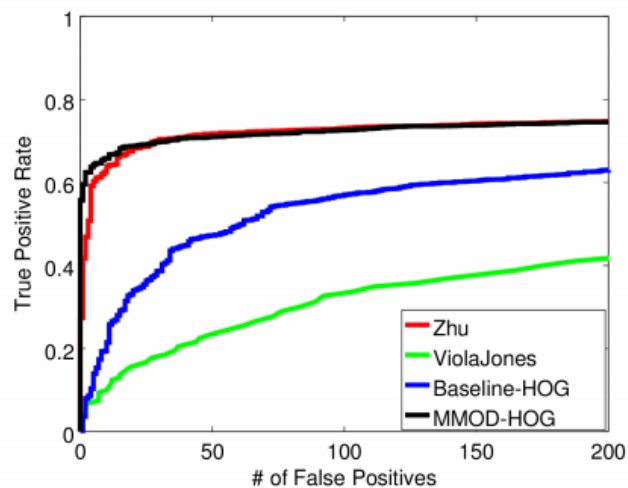
O método utilizado nesse trabalho é uma otimização da versão base, de (KING, 2015) consiste na ideia de otimizações na utilização das *sliding windows* que são usados nos classificadores binários, o desempenho de tal aplicação no Fddb pode ser observado na Figura 2.3.

Figura 2.2 – Exemplificação das etapas do Método HOG após transformação na imagem de entrada.



Fonte: Adaptado de Panceri et al. (2015).

Figura 2.3 – Desempenho do método escolhido no Fddb, o chamado MMOD-HOG em comparação ao HOG tradicional.



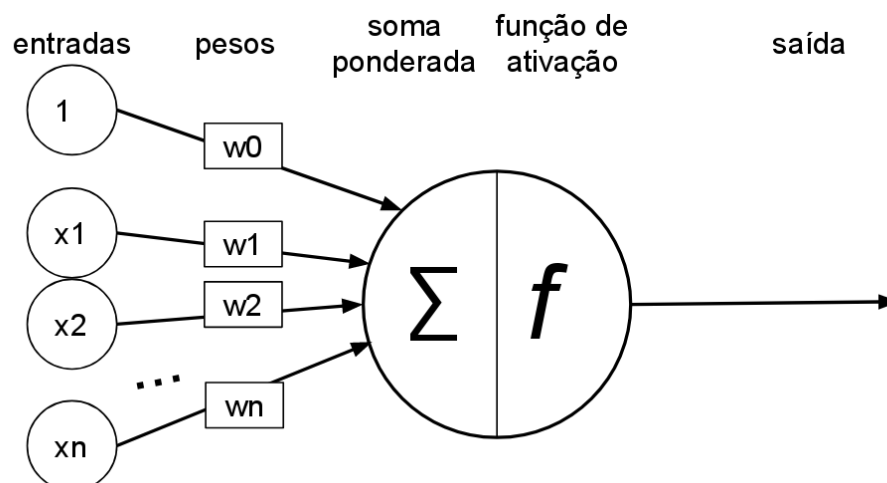
Fonte: Retirado de King (2015).

### 2.1.2 Redes Neurais

Segundo (VASCONCELOS; CLUA, 2017) sistemas que usam redes neurais são aqueles que aprendem padrões intrínsecos que constituem uma certa tarefa através da observação de grandes volumes de dados e tem a capacidade de generalizar os modelos aprendidos para encontrar a resposta esperada em novos casos, sendo que esses sistemas são aplicáveis nas mais diversas áreas, como na análise de sinais visuais com imagens e vídeos, sinais de áudio, de linguagem natural, entre outros tipos de sinais e além da combinação de sinais de natureza distintas.

O conhecimento sobre redes neurais é bastante antigo, desde que (ROSENBLATT, 1962) propôs o modelo de Perceptron (Figura 2.4), o modelo consiste em valores de entrada conhecidos e pesos que são ajustados conforme o processo de treinamento da rede ocorre.

Figura 2.4 – Representação clássica do modelo Perceptron de rede neurais.



Fonte: Retirado de Vasconcelos e Clua (2017).

Os dois elementos representados na Figura 2.4 e que devem ser levados em consideração, o somatório da imagem e a função de ativação, o somatório representa a multiplicação de todos os pesos  $w$  com a sua respectiva entrada, existe ainda um valor chamado de bias que também é multiplicado por um peso e participa do somatório.

O bias é um valor que tem como objetivo deslocar a função de ativação escolhida para que a classificação final ocorra, além disso se uma rede neural é desenvolvida sem o bias, qualquer que seja a sua classificação, na representação no espaço ela sempre passará pela origem, uma maneira fácil de imaginar é pensar em uma função afim de primeiro grau onde o coeficiente linear é zero.

A função de ativação é uma função que recebe um ou mais valores e gera um valor para aquele neurônio, geralmente ela tem o objetivo de transformar o problema em um não linear, pois o objetivo é aumentar o poder de expressão da rede, uma vez que sem

ela as redes neurais só resolveriam problemas de regressão linear, as primeiras funções desenvolvidas tinham esse objetivo.

### 2.1.3 Redes Neurais Convolucionais

As redes neurais convolucionais são um conceito conhecido desde o trabalho de (FUKUSHIMA, 1980) e vem se modernizando nas técnicas usadas nas camadas internas desde então, usa-se essas redes para analisar imagens, uma vez que redes neurais comuns não escalam de uma maneira aceitável sobre as mesmas, visto que uma imagem contém informação em três dimensões, sendo altura, largura e os canais de cor.

Seguindo a lógica para cada uma dessas dimensões seria necessário um peso associado na rede para cada *pixel*, isso geraria um número de parâmetros não processável para o ajuste no treinamento da mesma, a rede neural convolucional trabalha com os neurônios em matrizes de 3 dimensões e analisa parte dos *pixels*, esses são divididos em *grids*.

As redes neurais convolucionais representam uma arquitetura de aprendizagem profunda, que fazem parte dos métodos de aprendizado de máquina, a aprendizagem profunda se caracteriza no uso de múltiplas camadas para extrair características de uma entrada e sua estrutura permite o processamento da informação de uma maneira não linear.

Essas redes são formadas por um conjunto de camadas na qual cada uma exerce uma função sobre a entrada de *pixels*, os tipos de camadas mais comumente utilizadas são *Convolutional Layer*, *Pooling Layer*, e *Fully-Connected Layer*.

A primeira é a camada que na qual ocorre a aplicação de filtros sobre *grids* divididos da imagem de entrada, esses *grids* percorrem toda a entrada para gerar a nova entrada, a escolha dos valores do filtro faz com que certa característica da imagem de saída seja obtida, tais valores são obtidos por treinamento.

O nome convolução vem de o termo matemático de combinação de duas funções para produzir uma terceira, ou seja, faz a combinação de dois *sets* de informações, sendo que o filtro mencionado também é chamado de *kernel* e o resultado é chamado de *feature map*.

O termo *stride* é usado para mencionar quanto que o *grid* selecionado anda para o lado, uma técnica comum empregada é a chamada *padding* que coloca zeros na borda da matriz para não ocorrer redução de tamanho na operação.

A segunda é a camada nas quais ocorrem reduções do tamanho total da entrada, nota-se que ocorre perda de informação, porém a perda é feita de maneira que não atrapalhe a classificação final.

Existem diferentes funções que podem ser escolhidas na redução, a mais comumente utilizada é a *max pooling* que pega o maior valor de uma matriz de tamanho es-

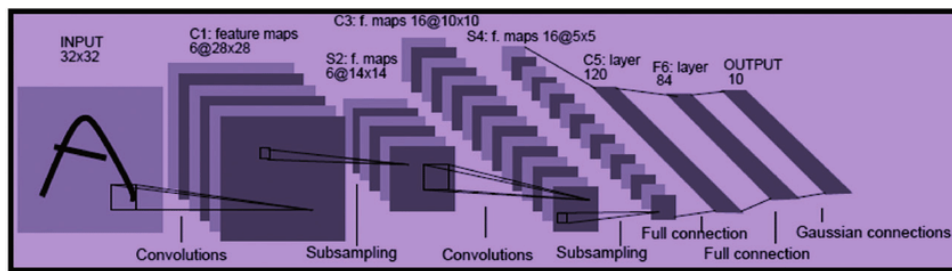


colhido, e passa para o resultado, aqui também existe um *stride* para percorrer a matriz maior.

A terceira é uma camada como as que existem em redes neurais comuns, ela é utilizada antes da classificação final, a inspeção do que os filtros obtiveram resultarão em uma classificação, que pode ser por se exemplo se existe faces naquela imagem ou não.

Um exemplo simples de como uma rede neural convolucional pode ser estruturada pode-se observar na Figura 2.5 na qual usa-se as três estruturas mencionadas, nota-se que podem ser realizadas várias combinações de camadas e ainda existem funções de ativações que são aplicadas depois de certas camadas.

Figura 2.5 – Representação de uma arquitetura de redes neurais convolucionais.



Fonte: Adaptado de Lecun et al. (1998).

#### 2.1.4 MTCNN

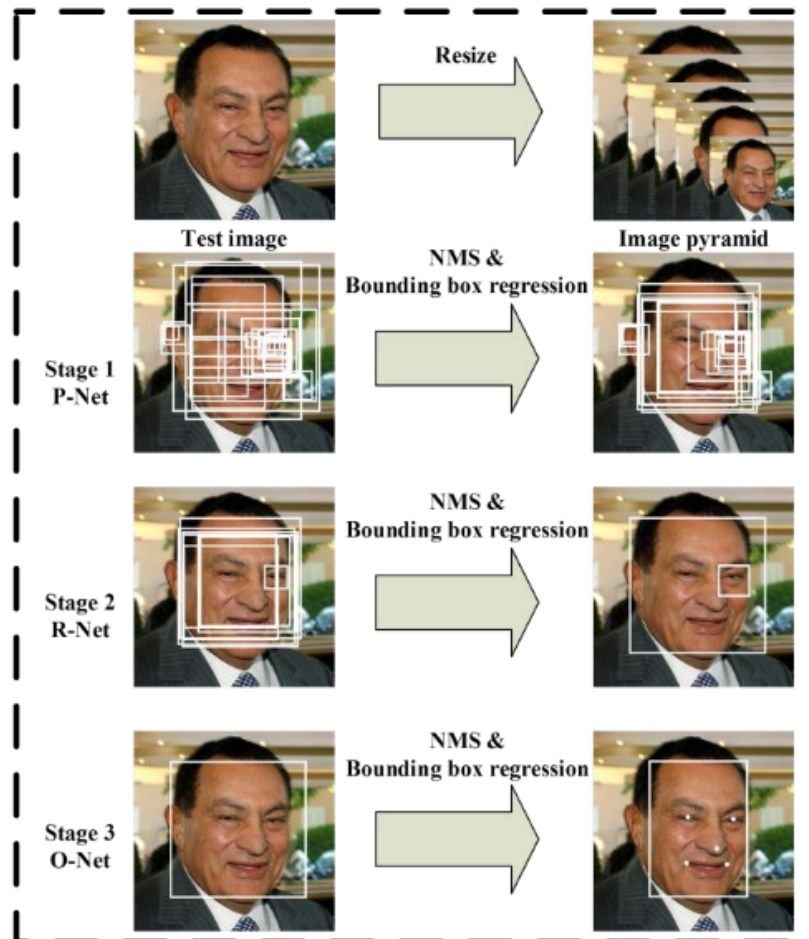
O MTCNN é um método de (ZHANG et al., 2016) que usa três redes neurais convolucionais para gerar uma Região de Interesse (ROI), além de obter os recursos que ajudam na detecção de faces, como olhos, boca e nariz, o MTCNN usa imagens RGB que tem três canais de cores.

O estágio de funcionamento da rede, após seu treinamento, como é utilizado nesse trabalho, ocorre da maneira descrita na Figura 2.6, na qual ocorre a criação de uma pirâmide com imagens de vários tamanhos para utilização do conceito de *sliding windows* que servem para varrer essas imagens a procura de possíveis classificações positivas, o processo depois da criação da pirâmide é dividido em três estágios:

1. A entrada é enviada a rede neural (P-Net) para a obtenção da janela de candidatos e dos vetores que representam as extremidades do objeto detectado, e ocorre também a utilização de um algoritmo de supressão não máxima para diminuir os candidatos que contém pontos em comum;
2. Ocorre a inserção dos candidatos em uma nova rede (R-Net) que rejeita mais candidatos e realização de nova supressão;

3. Nova inserção dos candidatos na rede (O-Net), sendo que essa rede envolve mais parâmetros para obtenção final dos candidatos selecionados, também obtém-se a posição das características em uma face.

Figura 2.6 – Representação de funcionamento do MTCNN e dos resultados entre sub-redes.

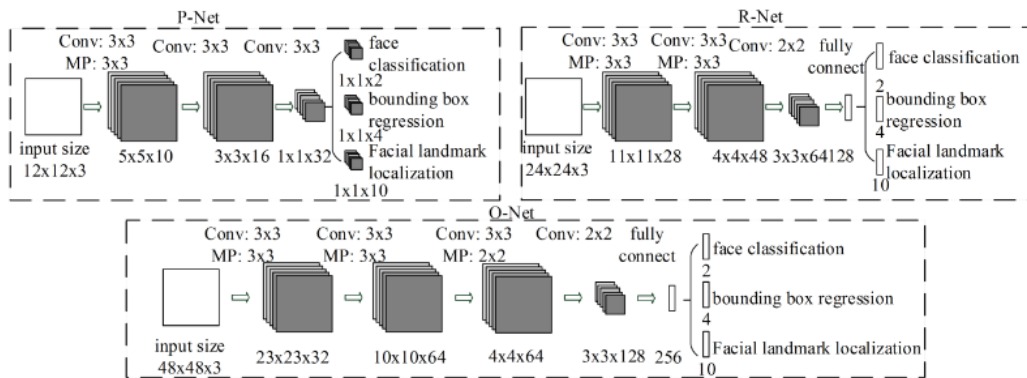


Fonte: Retirado de Zhang et al. (2016).

O treinamento da rede em si é com uso do método de *backpropagation* que calcula os valores do gradiente da função de perda (função que calcula o quão longe eu estou do meu objetivo no treinamento) com relação a cada peso e *bias* da rede.

Depois do *backpropagation* os pesos e *bias* são atualizados com o método do gradiente descendente estocástico, como o método deve ser treinado para realizar diferentes tarefas em uma única rede, é necessário utilizar mais de uma função de perda, pode-se observar a necessidade das várias funções de perda com a observação da última camada na arquitetura das redes na Figura 2.7.

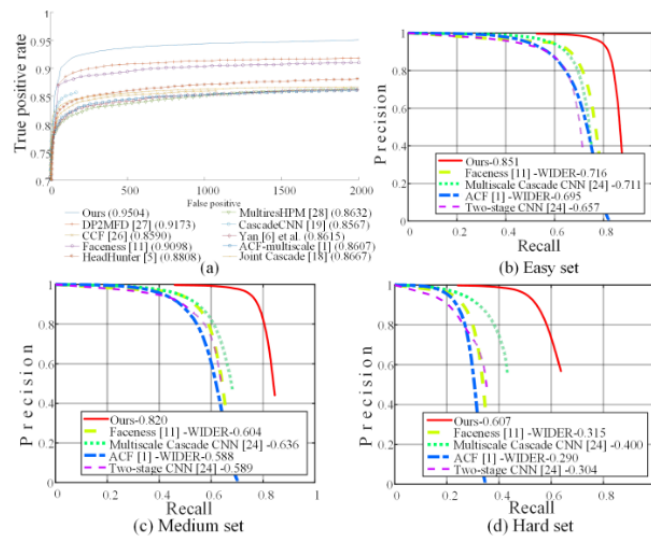
Figura 2.7 – Arquitetura do MTCNN.



Fonte: Retirado de Zhang et al. (2016).

Em termos de desempenho quando avaliado nos *databases*, pode-se verificar no Gráfico 2.1 e a Figura 2.8.

Figura 2.8 – Gráfico superior esquerdo representa o FDDB e os outros gráficos os diferentes *sets* do WIDER.



Fonte: Retirado de Zhang et al. (2016).

### 2.1.5 Métricas de Classificadores Binários

O que mede o sucesso de tais métodos, são métricas obtidas que são usadas para construir um gráfico, esse é conhecido como Característica de Operação do Receptor (ROC), o gráfico serve para ilustrar a capacidade de diagnóstico de um classificador binário.

rio, a construção da curva ROC depende da escolha das taxas após a coleta dos seguintes dados:

- verdadeiro-positivos
- falso-negativos
- verdadeiro-negativos
- falso-positivos

Um valor calculado é o Valor preditivo positivo ou *precision*, que serve para analisar a frequência que o classificador foi correto fazendo a predição da classe positiva, a fração é os valores de verdadeiro-positivos sobre verdadeiro-positivos somados aos falso-positivos (localizado um objeto, porém o objeto não existe), basicamente representa de todos os objetos encontrados quantos estavam certos.

Outro valor calculado é a taxa de verdadeiro-positivos ou *recall*, que serve para saber de todos os objetos que deveriam ter sido encontrados, quantos realmente foram, deve-se usar os valores de verdadeiro-positivos (objetos corretos encontrados) sobre os verdadeiro-positivos somado aos falso-negativos (objetos que não foram encontrados e deveriam ter sido).

Observa-se que se o *recall* for 1 todos os objetos que deveriam ter sido encontrados foram e não foi ignorado nenhum objeto (valores de falso-negativos), também nota-se que é possível o cálculo de outras taxas com a utilização dos valores de verdadeiro-negativos (objetos não encontrados quando ausentes).

O Fddb usa a métrica de *recall* e valores de falso-positivos para gerar suas curvas ROC, já o WIDER usa *recall* e *precision*, além disso depois que as métricas são coletadas das imagens, isso gera um ponto, logo é preciso obter mais pontos, isso é realizado, alterando-se a confiança de detecção da rede, logo se obtém os pontos para criar a curva.

Observa-se que no Fddb o tamanho mínimo de imagens para detecção das faces é 20x20 *pixels*, no WIDER existem diferentes *sets* de imagens de acordo com a dificuldade de detecção, tudo isso para serem gerados diferentes curvas ROC, para a comparação dos mesmos.

A comparação na curva ROC para o desempenho da rede nas classificações é basicamente à área abaixo da curva, quanto maior melhor, sendo que o ponto (1,0) seria o classificador ideal no qual não apresenta erros (assumindo um gráfico com *recall* no eixo y).

Destaca-se que nesse trabalho o foco é sobre imagens, quando os métodos são aplicados sobre vídeos, deve-se analisar se o algoritmo escolhido de detecção é rápido o suficiente para funcionar em tempo real.

## 2.2 MÉTODOS DE RECONHECIMENTO

Reconhecimentos de faces consiste na tarefa de verificar se uma ou um conjunto de faces pertencem a uma mesma pessoa, a partir da comparação entre as características obtidas das faces, o primeiro método que resolveu tal problema de maneira superior ao próprio reconhecimento humano foi (LU; TANG, 2014).

Desde então houve uma explosão na área em sistemas que têm a mesma capacidade, porém os melhores são geralmente comerciais como é o caso do sistema que utiliza redes neurais desenvolvido pela empresa IntelliVision<sup>1</sup>.

O objetivo dos métodos de reconhecimento consistem em extrair recursos das faces já detectadas e assim poder ser possível uma comparação das mesmas, o que não é trivial de se realizar uma vez que segundo (LU; TANG, 2014) o LFW contém imagens que têm variações de pose das faces, iluminação, expressão, etnicidade, idade, gênero, estilo de cabelo, e outros parâmetros.

Em (WANG; DENG, 2018), a análise dos métodos pode ser observada e seu desempenho no LFW além da arquitetura utilizada, nota-se que a maioria dos altos desempenhos de reconhecimento utilizam redes neurais residuais, a biblioteca Dlib utiliza uma rede desse tipo para realização do reconhecimento, logo ela foi escolhida para uso nesse trabalho.

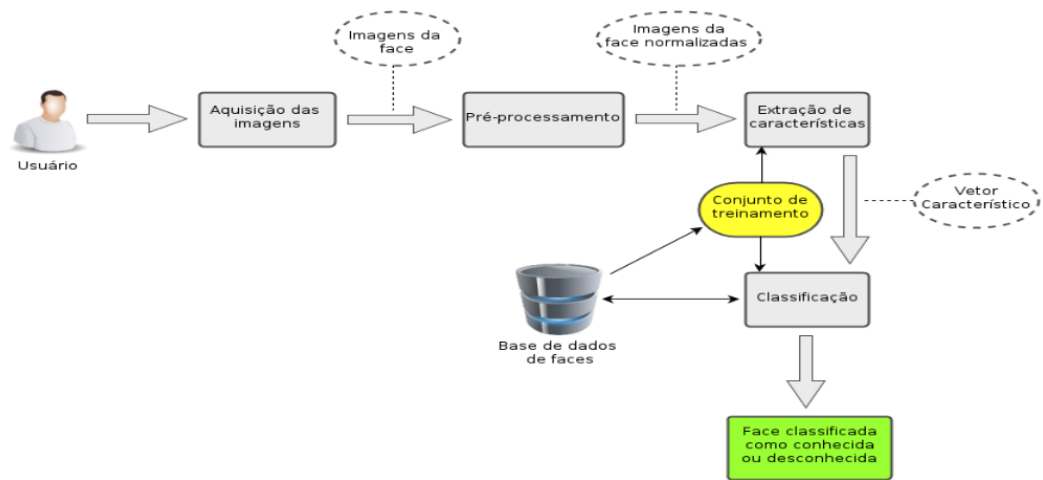
Nota-se que essa análise não conta com métodos que fazem a extração de características de uma face em três dimensões, logo esse tipo de avaliação representa o que melhor existe em duas dimensões.

O processo de reconhecimento consiste de basicamente quatro etapas, a detecção da face, a normalização da face para comparação, a extração de características e a comparação final, a Figura 2.9 exemplifica o processo.

---

<sup>1</sup><https://www.intelli-vision.com/facial-recognition/>

Figura 2.9 – Descrição do processo de Reconhecimento facial.



Fonte: Retirado de Diniz et al. (2013).

A normalização é um processo necessário, pois depois de serem detectadas as faces, elas não necessariamente terão as mesmas dimensões e alinhamento, então é necessário aplicar uma escala e alinhamento da face através de alguma característica da mesma.

Nota-se que existem outras operações que podem ser realizada em imagens antes do reconhecimento como equalização do histograma de uma imagem, no qual foi realizado em (PANCERI et al., 2015), porém observou-se que a melhora do desempenho depende do método utilizado.

O pré-processamento de imagens pode melhorar o desempenho do reconhecimento como mostra (BARNOUTI, 2016), porém deve-se tomar cuidado em fatores como escala, pois a imagem pode perder toda a informação dependendo do fator escolhido.

### 2.2.1 Reconhecimento na Dlib

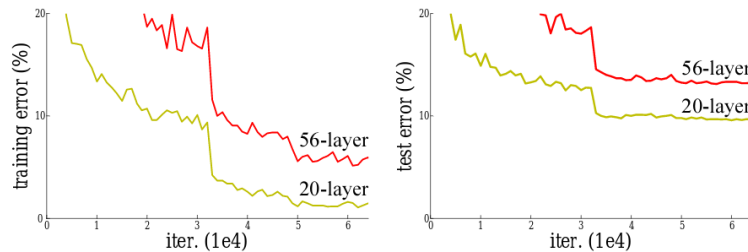
O reconhecimento facial na biblioteca Dlib como já citado possui um desempenho acima do reconhecimento humano, porém a versão mais eficiente em termos de tempo de execução possui um desempenho de 99,13% no LFW, essa versão não faz alterações de pré-processamento de uma face, a com taxa de reconhecimento maior faz.

A rede utilizada nesse trabalho é um modelo com os pesos das camadas previamente treinados, o treinamento possuiu um número de imagens com 3 milhões de faces e 7485 indivíduos diferentes, além de 29 camadas de convolução na rede.

O modelo de rede utilizado é uma adaptação de (HE et al., 2016), no qual é descrito o problema que ocorre quando o número de camadas de uma rede profunda aumenta e,

após o treino seu desempenho em termos de precisão na classificação piora, os autores denominam o problema de degradação, pode-se observar o problema na Figura 2.10 .

Figura 2.10 – Erro de treinamento (esquerda) e erro de teste (direita) em redes com 20 e 56 camadas.



Fonte: Retirado de He et al. (2016).

O problema ocorre devido à atualização dos valores de pesos da rede não ocorrer corretamente pois os valores calculados de gradientes são muito pequenos, a solução encontrada são as chamadas redes neurais residuais, nas quais são inseridos conexões na rede para a informação de camadas anteriores poder ser transmitida na rede, nas redes tradicionais uma camada só recebe os valores da camada anterior e passa sua saída para a próxima, nas residuais existe o chamado resíduo.

O uso de tais redes tornou possível a melhora de desempenho da mesma após o processo de treinamento com a utilização de mais camadas, a versão usada nesse trabalho apresenta desempenho similar ao de uma rede residual com 34 camadas descrita no artigo já citado.

## 2.2.2 Fatores Limitantes

Quando fala-se de câmeras que coletam imagens para reconhecimento, fatores como a quantidade de *pixels* são importantes pois os mesmos são analisados, imagens de baixa resolução (número baixo de *pixels* por polegada) sofrem com a tarefa de reconhecimento segundo (XU; LIU; LI, 2014).

Nota-se também que é mais fácil fazer o chamado *downsampling* de uma imagem, mesmo que ocorra perda da informação do que tentar aumentar a quantidade *pixels* de uma imagem, pois esse processo requer criar *pixels* e isso geralmente resulta em imagens borradas.

A iluminação do ambiente é um dos fatores que mais têm influência no reconhecimento de faces segundo (BRAJE et al., 1998) é extremamente complexo de entender como afeta uma imagem visto que a posição da fonte luminosa gera diferentes sombras e

outros resultados em uma imagem, o ambiente instalado da câmera apresenta variações de iluminação que dependem da hora do dia.

A câmera tira fotos a partir do movimento de pessoas, quando fotos são tiradas pelo sensor pode ocorrer imagens com o chamado *Motion blur* que é a pessoa estar em movimento durante o tempo de exposição da foto que causa um borrão na foto, o que dificulta a tarefa de reconhecimento.

A variação da pose também é um fator que interfere na porcentagem total de reconhecimento, uma vez que fica difícil comparar imagens em várias rotações da face, a oclusão é um fator chave também visto que uma face que não aparece por completo não pode ser reconhecida.

O ruído também é um fator a ser considerado visto que as fotos podem não conter o tempo correto de exposição devido à mudanças na iluminações do ambiente, gerando assim imagens com valores na intensidade dos *pixels* alterados.

### 2.2.3 Trabalhos Relacionados

No trabalho de (PRADO, 2017) são realizados experimentos com diversos métodos em diversas configurações para se analisar se é possível criar um sistema reconhecimento facial em uma academia, o autor concluiu que as imagens obtidas e as tentativas de reconhecimento não apresentavam um valor aceitável para a criação do sistema, os métodos utilizados não usavam redes neurais convolucionais.

No trabalho de (XU; LIU; LI, 2014) o objetivo consiste de melhorar o desempenho de reconhecimento a partir de imagens de câmeras de segurança, os autores avaliam que existem três fatores principais que afetam o desempenho do reconhecimento:

1. Qualidade da imagem obtida;
2. Tipo da câmera;
3. Distância entre a câmera e a face.

De acordo com seus experimentos quanto maior a resolução de uma imagem, maior a capacidade de reconhecimento, nota-se que esse trabalho não menciona redes neurais convolucionais nos métodos escolhidos para testes.

O uso de reconhecimento pode ser aplicado em casos como (DINIZ et al., 2013), nas quais foram aplicados métodos menos robustos que os escolhidos nesse trabalho, no contexto de identificar alunos através da *webcam* de computadores, e o desempenho de reconhecimentos ficou acima de 90%.

O trabalho de (CORVEE; BAK; BREMOND, 2012) é um sistema que utiliza de duas câmeras de segurança para detecção e re-identificação de faces, ele realiza o treinamento



das detecções e reconhecimento usando características locais de uma imagem e faz uso também de árvores de decisão, que são modelos estatísticos para previsão e classificação de dados.

Árvores de decisão sozinhas tem um baixo poder preditivo e para a combinação de diversas árvores usa-se o chamado Adaboost que segundo (WOZNIAK; GRAÑA; CORCHADO, 2014) é um algoritmo que tem o objetivo de obter um classificador com alto poder preditivo.

Nesse trabalho são somente analisadas fotos maiores que 50x50 *pixels*, ignorando assim problemas de resolução baixa, a ideia é utilizar uma câmera para coletar um número de imagens e ocorrer um processamento das características da pessoa e então colocar os dados em um banco e a outra câmera tenta o reconhecimento.

Destaca-se também que o treinamento dos classificadores foi usando dados diferentes daqueles que foram testados, e que os autores usam diversos métodos conectados para o reconhecimento das pessoas.

Oteve-se 85% de reconhecimento quando aplicado em um simples banco de imagens para teste, com variação de iluminação, pose e expressão.

Nota-se que usando as imagens para testes elas foram divididas e simulou-se um número de imagens para extração das informações e outro número para reconhecimento, com o *dataset* criado por eles os resultados foram melhores tendo 91% na primeira câmera (também tentava reconhecer) e 100% na segunda, ressaltou-se que certos indivíduos passaram muito rápido pela câmera e não ocorreu extrações suficientes.

Sobre o reconhecimento facial em (DELBIAGGIO, 2017) foi feita a comparação com a biblioteca OpenFace de (AMOS; LUDWICZUK; SATYANARAYANAN, 2016) que tem como base a rede FaceNet de (SCHROFF; KALENICHENKO; PHILBIN, 2015) com o métodos que não usam redes neurais, a conclusão da comparação é que a rede neural consegue se sair melhor quando as imagens que são usadas para comparação mudam de ambiente, devido a sua grande capacidade de generalização.

Na detecção com alto desempenho pode se observar os métodos de estado da arte em (ZHANG et al., 2019), nos quais mesmo com os parâmetros de *precision* e *recall* colocados no máximo não se obtém um classificador perfeito, isso ainda contando os testes em *datasets* considerados fáceis.

Sobre as bases de imagens utilizadas para o reconhecimento é importante notar segundo (SILVA; CINTRA, 2015) que a diferença entre os conjuntos de imagem podem variar em:

- **A** - As imagens foram obtidas em diferentes ambientes, variando o plano de fundo das imagens;
- **E** - As imagens apresentam faces com expressões, tais como alegria, tristeza ou surpresa;

- **I** - As imagens de face apresentam variação de intensidade e ângulo da iluminação;
- **O** - As imagens apresentam algum tipo de obstrução no rosto do indivíduo, tais como óculos, cachecol ou chapéu;
- **P** - As imagens apresentam o rosto do indivíduo em posições diferentes (cabeça inclinada para um dos lados, abaixada ou levantada);
- **T** - As imagens foram obtidas com diferença de alguns dias ou meses, podendo apresentar características de envelhecimento como crescimento ou corte de cabelo e/ou barba.

### 3 SISTEMA DE DETECÇÃO E RECONHECIMENTO DESENVOLVIDO

A aplicação desenvolvida permite a detecção de pessoas de acordo com fotos, sendo as escolhas do método de detecção parametrizadas, além de escolhas opcionais de pré-processamento.

Sobre a detecção a aplicação realiza a contagem da hora em que um elemento ou mais elementos foram detectados e se armazena esses dados, sendo que a visualização dos dados é disponível de maneira geral ou individual pelo sistema.

O funcionamento da aplicação é por dias ou vários dias, tanto para a detecção como para o módulo de reconhecimento, esse módulo consta com a parametrização no sentido de ser escolhida uma distância máxima para a comparação de faces.

Ainda no módulo de reconhecimento destaca-se a capacidade da visualização de usuários quando acontece um reconhecimento, os usuários desconhecidos e também a capacidade de ver as distâncias entre tentativas de reconhecimento.

O capítulo atual descreve o desenvolvimento da aplicação web por meio das bibliotecas utilizadas, diagramas da *Unified Modeling Language* (UML), as capacidades do sistema, além de explicar especificações necessárias para o entendimento do sistema de uma maneira geral.

#### 3.1 CASOS DE USO

Diversos métodos podem ser empregados para a descrição de um sistema segundo (PATEL P. E.; PATIL, 2013) a norma que se tornou padrão para a indústria de desenvolvimento de códigos orientado a objetos é a UML, que é uma linguagem formada por um conjunto de diagramas que representam os aspectos necessários na descrição de um *software*.

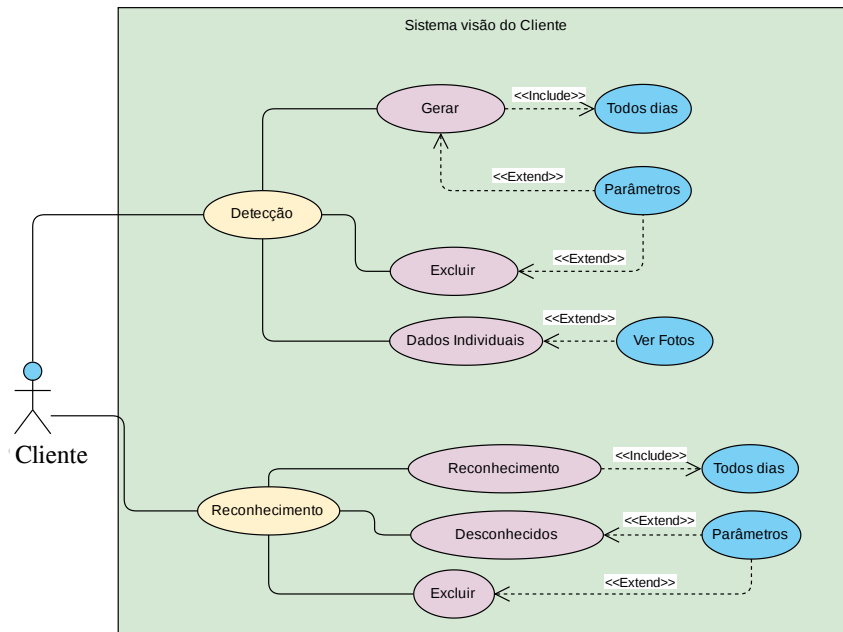
Um desses diagramas é o diagrama de caso de uso, (AL-FEDAGHI; ALRASHED, 2013) descreve que no desenvolvimento de requerimentos de usuários o diagrama de caso de uso é utilizado em 72% dos casos de desenvolvimento de requisitos do usuário em sistemas, porém somente esse diagrama não é suficiente são necessários outros diagramas.

Segundo (CONSTANTINE; LOCKWOOD, 2001) o diagrama de casos de uso auxiliam designers de interfaces de usuários, pois o diagrama ajuda o designer a ter um entendimento das necessidades do usuário, e também ajuda os engenheiros de *software*, visto que os mesmos têm a visualização de como implementar os requisitos funcionais.

O trabalho atual consiste no desenvolvimento de uma aplicação web, pode se observar na Figura 3.1 as funcionalidades básicas que um usuário possui e nota-se que no caso do diagrama citado foi somente avaliado o lado do cliente, como o a aplicação é

web, cada ação desse diagrama é uma requisição mandada ao servidor com o *Hypertext Transfer Protocol* (HTTP), o servidor então realiza ações e retorna uma resposta ao cliente.

Figura 3.1 – Diagrama de casos de uso representando as ações de detectar e reconhecer faces.

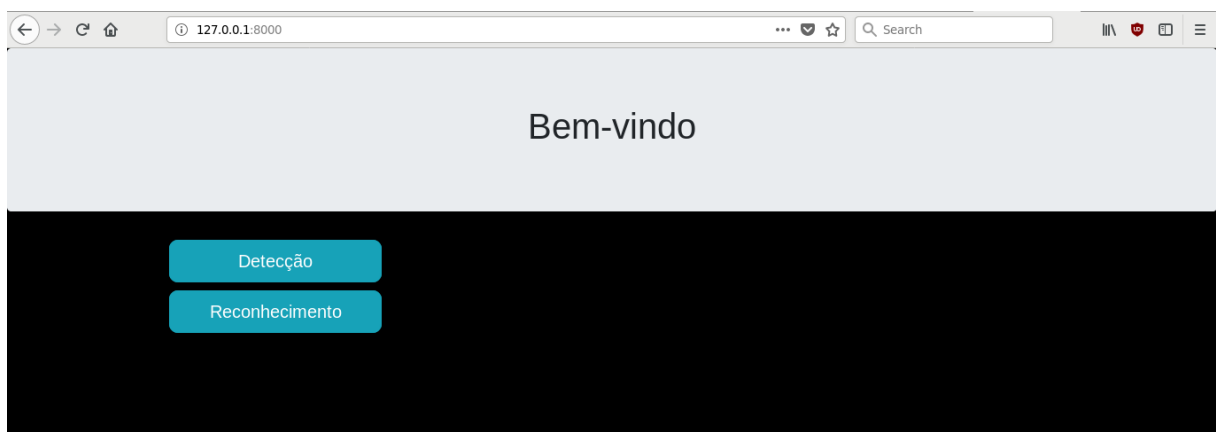


Fonte: Autor.

O diagrama da Figura 3.1 representa as interações que um cliente tem ao se deparar com o sistema, pode ser visualizado na Figura 3.2, e as funcionalidades são:

- A detecção (Capítulo 3.2) é a ação que proporciona o usuário realizar a detecção em fotos separadas por dias, com escolha de parâmetros.
- O reconhecimento (Capítulo 3.3) representa a parte do sistema em que um usuário pode tentar reconhecer as faces detectadas em um dia ou vários, além da escolha de parâmetros, sendo as faces então comparadas com as existentes no banco de dados.

Figura 3.2 – Interface inicial da aplicação.



## 3.2 DETECÇÃO FACIAL

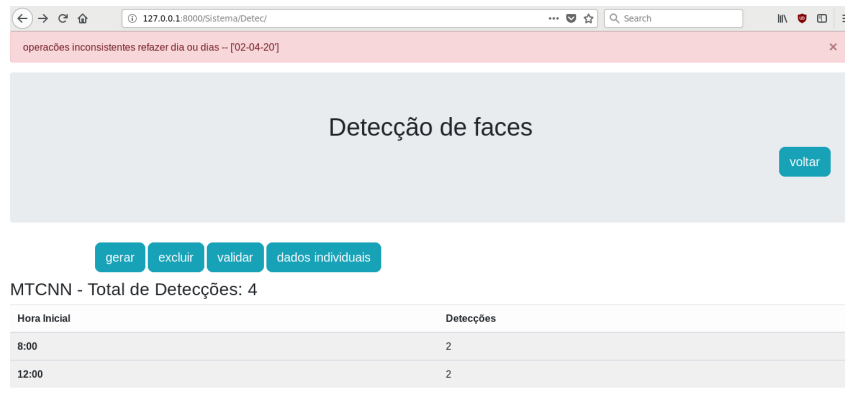
A ação de detectar faces é um dos casos de usos, a tela que o usuário interage é a Figura 3.3 e pode-se observar quatro botões de controle, um botão para detectar faces, um botão para excluir faces, um botão extra para demonstrar a consistência dos dados, e um botão que permite ver as informações separadas por dias e as fotos de cada dia.

A ideia de gerar ou detectar faces funciona por dias, sendo que as fotos das câmeras utilizadas são colocadas em pastas separadas por dia, logo se desenvolveu a detecção sobre as mesmas, a Figura 3.4 ilustra a opção *default*, que detecta todas as fotos de todos os dias, ou a opção de escolher um dia.

A tela de configurações representa todas as escolhas que um usuário possui na manipulação da imagem ao longo da detecção, sendo os parâmetros a seguir variáveis e os últimos quatro itens opções de pré-processamento:

- **Dias:** Escolha dos dias para a ação de detecção ocorrer;
- **Tamanho da foto:** Escolha do tamanho que ele deseja da foto se usado a escala da mesma;
- **Threshold:** Se possível a variação do desempenho do método escolhido;
- **Método de Detecção:** O usuário escolhe qual método ele vai usar;
- **Arquivo com nome para testes:** O usuário escolhe se ele vai usar fotos com o tempo atual ou o nome do arquivo com a hora na operação;
- **Fotos em Bordas:** Utilizar fotos que estão cortadas nas bordas depois de detectada uma face;
- **Rotação:** Utilização da rotação a partir da posição dos olhos de um indivíduo;
- **Escala:** Utilização da escala ou não;
- **Método para escala:** Variação do Método utilizado para escala.

Figura 3.3 – Representação da tela de detecção com um erro de validação.



Fonte: Autor.

Figura 3.4 – Tela de gerência de configurações para detecção.

Digite uma data ou default para todas datas:

default dd-mm-yyyy com os traços

70 Tamanho da foto em px

0.7 Threshold

Método de Detecção

MTCNN

Arquivo com nome sem HORA (testes)

Pré-Processamento:

Utilizar faces fora das Bordas

Utilizar rotação

Utilizar escala

Método de interpolação da escala:

INTER\_LINEAR

executar

Fonte: Autor.

O princípio da tela que gera as exclusões é o mesmo da tela que gerencia as detecções, sendo que a lógica é a exclusão de uma data ou todas as datas, sendo que o sistema só exclui as fotos de faces que foram geradas pelo próprio.

Nota-se na Figura 3.3 que são gerados dados relativos ao intervalo de hora em que

uma pessoa foi detectada, nessa imagem ocorre a demonstração total dos valores sendo possível acessar os valores separadamente como mostra a Figura 3.5.

Figura 3.5 – Tela de dados individuais de detecção.

MTCNN - Fotos

- Data: 02-04-2019

Hora Inicial	Detecções
16:00	1

Data: 01-04-2019

Hora Inicial	Detecções
8:00	2
12:00	2

HOG - Fotos

- Data: 02-04-2019

Hora Inicial	Detecções
16:00	1

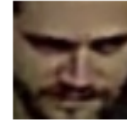
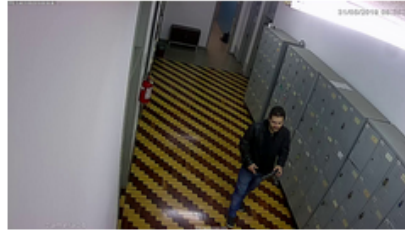
Fonte: Autor.

A Figura 3.6 demonstra uma parte da tela na qual o usuário pode ver as fotos e sub-fotos geradas pelo método de sua escolha, nota-se que todas as telas funcionam para mais de um método, nessa tela ainda pode-se observar o nome do arquivo que nesse caso é a hora, mas poderia ser nomes de set para testes sendo mostrados de uma forma ordenada.

Figura 3.6 – Exemplo de visualização das fotos e suas detecções.

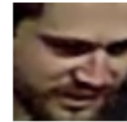
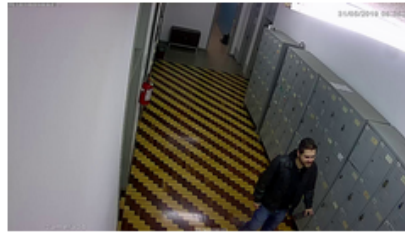
- Data - 01-04-2019

08:56:06



Confiança  
0.9912

08:56:07



Confiança  
0.9999

09:09:09



Confiança  
0.9994

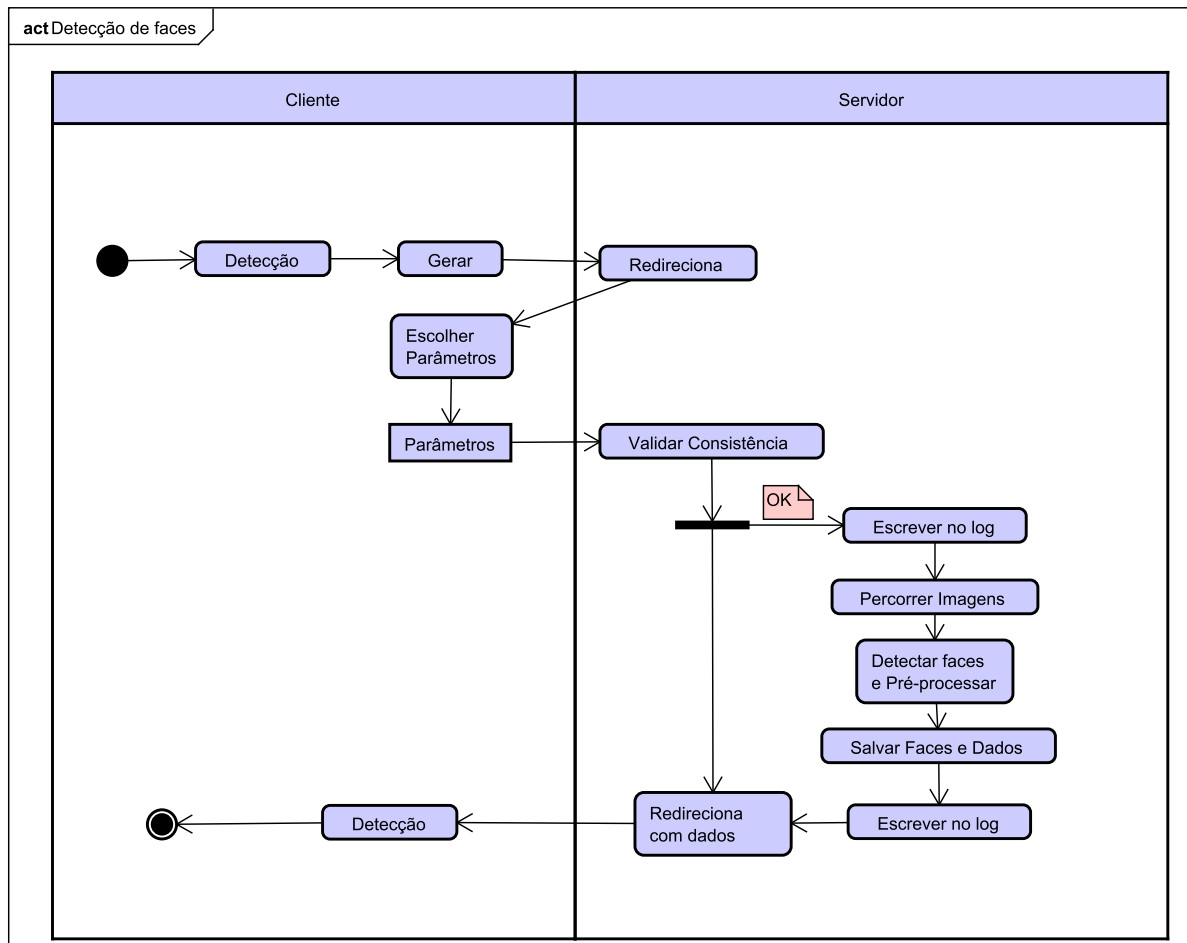
Fonte: Autor.

### 3.2.1 Detalhes Específicos

Pode-se observar o diagrama de atividades na Figura 3.7, no qual é descrito o processo de funcionamento da detecção das faces assumindo um dia de execução.



Figura 3.7 – Diagrama de atividades para detecção.



Fonte: Autor.

Analisando mais profundamente a execução no lado do servidor pode-se avaliar a lógica seguida como a seguinte:

- Escrever registro de início de operações no *log*;
- Percorrer fotos;
- Para cada foto detectar as faces usando um dos métodos explicados nesse trabalho;
- Para cada foto pré-processar a imagem do modo escolhido;
- Para cada foto salvar a face nos arquivos e dados utilizados no banco de dados;
- Escrever registro de fim de operações no *log*.

Pode-se ainda detalhar o funcionamento da etapa do pré-processamento da seguinte maneira, sendo esses itens opcionais:

- Descartar fotos que tenham ROI fora da imagem;

- Obter inclinação entre posição dos olhos e o eixo x;
- Obter ponto central da ROI;
- Rotacionar toda imagem;
- Usar o ponto central da ROI na matriz de rotação gerada;
- A partir do novo ponto central calcular a nova ROI;
- Recortar imagem;
- Escalar a imagem pela entrada do usuário;
- Coletar hora pelo nome da imagem ou hora atual e salvar os dados.

Nota-se que é importante notar que em todas essas etapas caso aconteça algum comportamento inesperado ocorre o redirecionamento com os dados para informar o cliente do ocorrido, na Figura 3.7 o tratamento de erros foi simplificado, pois o foco é mostrar o processo de detecção.

### 3.3 RECONHECIMENTO FACIAL

Outra função do caso de uso é a capacidade de reconhecimento facial, essa função pode ser vista na Figura 3.8, a descrição do sistema é basicamente coletar a descrição de faces através da rede neural já treinada, colocar esses valores no banco, e então compara-se novos dados de faces com os dados do banco.

A comparação entre faces ocorre, e se a distância for menor que a *default* ou a estipulada pelo próprio usuário, ocorrerá um *match*, como pode ser observado na Figura 3.8, a imagem mostra a face analisada e com quem ocorreu o *match*, a distância real e a do limite estipulado, além de hora e data do ocorrido, também mostra o total de pessoas reconhecidas e as pessoas desconhecidas (dados que estão no banco).

A imagem exemplifica as funcionalidades que um usuário pode utilizar, sendo possível reconhecer faces de um dia, ou vários dias, a exclusão das informações desejadas, ou ainda a capacidade de visualizar os usuários que tiveram suas informações salvas no banco, e quantos *matches* aquela face teve, essa funcionalidade pode ser observada na Figura 3.9.

Figura 3.8 – Tela de reconhecimento com botões.

Reconhecimento de faces - Dlib

[voltar](#)

Total de Reconhecidos	Total de Desconhecidos
2	5

[Reconhecimento](#)
[Desconhecidos](#)
[Excluir](#)

- Data - 02-04-2019

Hora Detec	Foto analisada	Foto banco	Distância Max	Distância Real	Detec Origem
4:12 p.m.			0.6	0.55	MTCNN

Fonte: Autor.

Figura 3.9 – Tela de Desconhecidos (faces no banco).

Fotos de desconhecidos	Matches	Distâncias Reais	Data
	1	<a href="#">Distâncias</a>	01-04-2019
	0	<a href="#">Distâncias</a>	01-04-2019
	0	<a href="#">Distâncias</a>	01-04-2019

Fonte: Autor.

Ocorre também na Figura 3.9 a possibilidade de o usuário verificar as tentativas de reconhecimento que ocorreram para uma face ser considerada desconhecida, o *link* nessa imagem redireciona o usuário para a Figura 3.10, na qual se observa as tentativas de reconhecimento executadas por uma face e o resultado do cálculo da distância.

Figura 3.10 – Tela de distâncias de tentativas de reconhecimento.

Foto do desconhecido	Foto da tentativa	Distância real
		0.64738791484047
		0.775459500107876
		0.752353898072336

Fonte: Autor.

Resumindo, o processo todo consiste em guardar os registros dos usuários que foram reconhecidos, e também de cada desconhecido que é inserido no banco, e as distâncias que levaram uma face a ser considerada uma desconhecida, as telas permitem a visualização das informações.

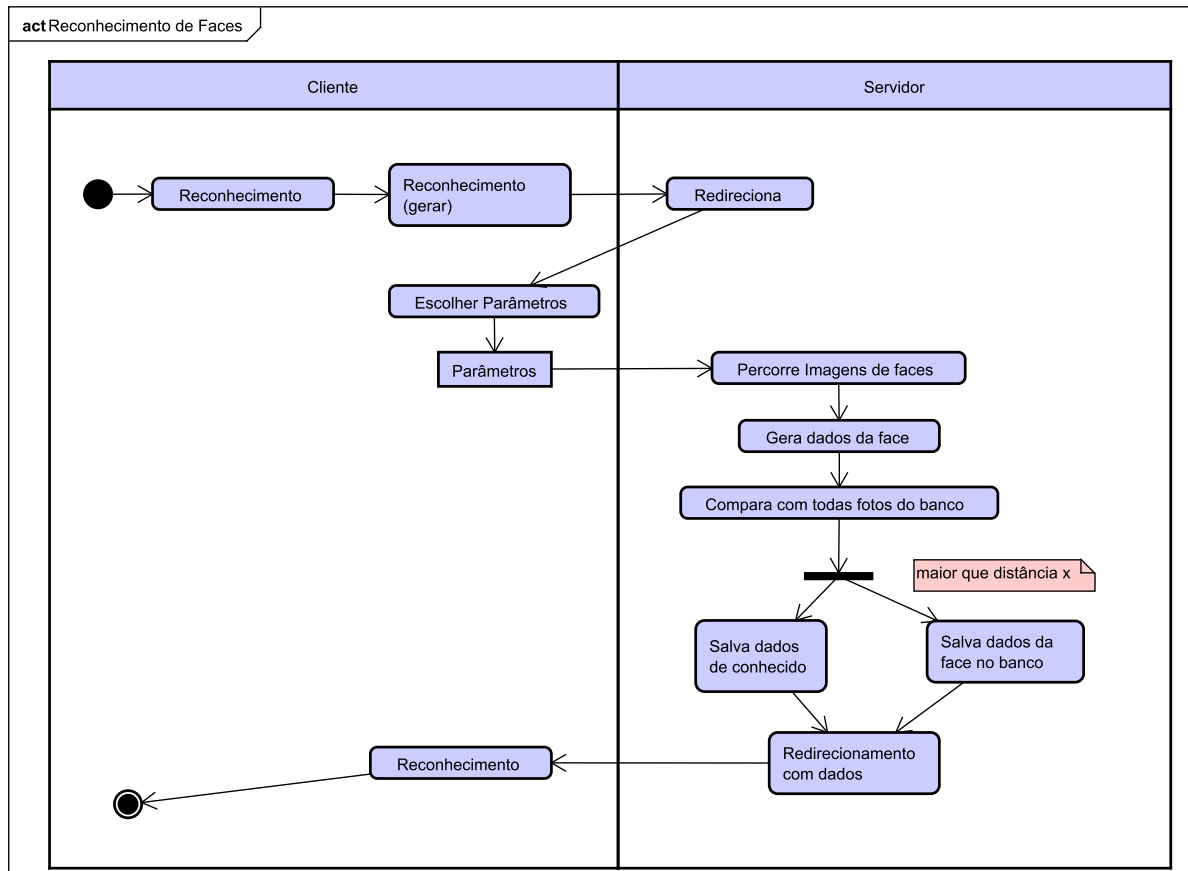
Além disso, os dados são gerados quando se deseja fazer a comparação entre os dados do banco e as faces que foram detectadas, sendo possível escolher parâmetros como o dia, também não ocorre a tentativa de reconhecimento de imagens que já foram reconhecidas anteriormente.

O diagrama de atividades do reconhecimento pode ser avaliado na Figura 3.11 e representa as atividades e testes realizados por cliente e servidor, analisando mais especificamente o lado de operações executadas no lado do servidor para uma face, a ideia é:

- Gerar os dados que descrevem a face;
- Comparar os dados gerados com os armazenados em um banco, usado distância euclidiana;
- Se conhecido, preencher o banco para visualização dos conhecidos, e somar um contador de quem sofreu o *match*;
- Se desconhecido, inserir no banco os seus dados para futuras comparações ou sua visualização.

Nota-se que o próprio sistema pode inserir os dados das faces no banco, ou pode se inserir previamente se assim desejado para execução testes de métodos.

Figura 3.11 – Diagrama de atividades para reconhecimento.



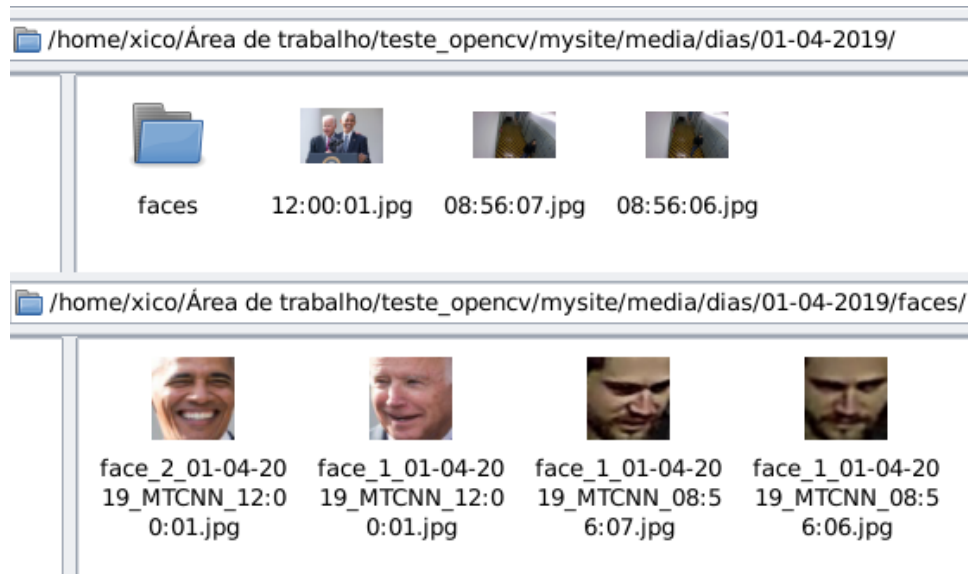
Fonte: Autor.

### 3.4 ARMAZENAMENTO DE FOTOS E FACES

A lógica para os diretórios de fotos é que existem os diretórios dos dias, nos quais as fotos estão inseridas, as fotos por si só apresentam a hora no seu nome para a coleta das informações, ou a aplicação também funciona com qualquer nome para os arquivos se desejado.

A Figura 3.12 representa um exemplo de um dia com as suas fotos, na qual as faces são colocadas em uma pasta filha chamada faces quando detectadas pelo sistema.

Figura 3.12 – Representação do diretório de um dia e das faces armazenadas pelo sistema no dia.



Fonte: Autor.

### 3.5 DETALHES EXTRAS DE IMPLEMENTAÇÃO

O sistema não se preocupa com as fotos serem transferidas pelo mesmo, é assumido que se tem acesso ao servidor ou alguma maneira de passar as fotos ao servidor, essa interface não manipula isso.

Considerando o problema da consistência de informações que podem ocorrer devido à interrupção da execução, na detecção foi utilizado a estratégia de escrever um *log* antes de realizar a operação, estratégia que banco de dados usam e que é mencionado em (JHINGRAN; KHEDKAR, 1992) como sendo a mais popular e comumente utilizada.

Como existe um *log* para as operações é possível validar se as informações estão consistentes ou não, a validação pode ser realizada de forma manual, ou quando a operação de detectar faces é chamada, ela ocorre para liberar o uso do processamento, a Figura 3.3 demonstra que a validação pode ocorrer com o uso de um botão e demonstra a mensagem de erro de uma operação, basta excluir o dia específico que ocasionou o problema e o programa estará consistente.

A consistência dos dados nas etapas de reconhecimento foram tratados com a utilização da funcionalidade do banco de dados de realizar operações atômicas no banco de dados, evitando assim o problema de inconsistência gerado na manipulação dos dados e problemas de execução.

### 3.6 BIBLIOTECAS UTILIZADAS

Para o desenvolvimento do presente trabalho foram utilizadas diversas bibliotecas de caráter *open-source*, a escolha foi realizada visando facilitar o tempo de implementação necessário e a integração entre as diversas funcionalidades, logo escolheu-se trabalhar com uma só linguagem, utilizando assim a linguagem de programação *Python* na versão 3.5.3.

A biblioteca *Open source Computer Vision (OpenCV)*<sup>1</sup>, na versão 4.0, foi utilizada para a manipulação das imagens e a realização das operações sobre as mesmas, além da sua implementação base do método de reconhecimento.

Para a parte de desenvolvimento web utilizou-se o *framework Django*<sup>2</sup>, na versão 2.2, que tem como objetivo facilitar a implementação web através de uma arquitetura definida em *model-view-template*, que é um modelo similar ao *model-view-controller* porém com a aplicação para web.

Sobre o Django ainda pode-se notar que a ideia do *framework* é facilitar a troca de mensagens que ocorrem entre cliente e servidor, além de auxiliar na relação do *Uniform Resource Locator (URL)* e as chamadas de funções do lado do servidor.

A biblioteca *Face Recognition*<sup>3</sup>, que facilita o uso da já citada *Dlib* ao longo do trabalho, e o banco de dados utilizado é o *PostgreSQL* na versão 9.6.

O último conjunto de bibliotecas é o do método de detecção *MTCNN*<sup>4</sup>, e as as bibliotecas de redes neurais que esse utiliza, no caso o chamado *Tensorflow*.

A implementação pode ser observada no *Github*<sup>5</sup>, a aplicação para ambientes locais pode ser acessada através de um *Docker*<sup>6</sup>, esse é um serviço que utiliza da virtualização do sistema operacional e o uso de contêineres que contam com as bibliotecas da aplicação.

---

<sup>1</sup><https://opencv.org>

<sup>2</sup><https://www.djangoproject.com>

<sup>3</sup>[https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)

<sup>4</sup><https://github.com/ipazc/mtcnn>

<sup>5</sup><https://github.com/ftfajardo/TCC>

<sup>6</sup><https://www.docker.com/>

## 4 ANÁLISES

Existem dois fatores que se destacam e que são estudados e analisados, são eles:

- **Desempenho do método de detecção:** Analisar os resultados das métricas obtidas de detecção nos diversos *sets*, de acordo com imagens coletadas de dois bancos de dados para detecção e imagens coletadas da câmera de segurança.
- **Desempenho sobre o reconhecimento:** Analisar o desempenho do reconhecimento com um banco de dados de imagens que contém várias pessoas expondo se o sistema consegue reconhecer elas e o desempenho sobre as imagens da câmera de segurança.

### 4.1 EXPERIMENTOS EM BASES DE IMAGENS PRONTAS

Para a escolha dos testes de detecção foram escolhidos duas bases de imagens com variações nos ângulos e uma com o adicional de variação na iluminação, sendo que ambas contém imagens de alta resolução para as faces.

Para os testes de reconhecimento escolheu-se uma base de imagens que contém várias pessoas, variação da iluminação e ângulo das faces.

#### 4.1.1 Testes de Detecção em bases de imagens prontas

Um banco de dados de detecção utilizado no trabalho é de (GOURIER; HALL; CROWLEY, 2004) e existem imagens de várias pessoas individualmente, uma pessoa tem suas fotos com variações de ângulo, nota-se que ocorre a variação de ângulo nos eixos vertical e horizontal.

O total de imagens são 186 para uma pessoa, no caso são dois *sets* com 93 imagens tiradas nas mesmas posições, uma com óculos e outra sem, os resultados podem ser visualizados no Quadro 4.1:

Quadro 4.1 – Métricas obtidas no banco de imagens com variação de ângulo na face de uma pessoa.

Método	Verdadeiro-positivos	Falso-negativos	Falso-positivos
HOG	104	82	0
MTCNN	184	2	1

Fonte: Autor.



Os métodos foram executados com os parâmetros de detecção para a maior detecção de elementos corretos, a *sliding window* que é usada para percorrer as imagens e obter as possíveis faces finais foram utilizadas nos parâmetros é de 40x40 *pixels*.

O desempenho quando avaliado sobre um contexto de um banco com variação da iluminação como é o caso do banco estendido de faces da Yale de (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001), que contém 586 imagens com variações de iluminação e poses para cada pessoa do *dataset*, seus valores são representados pelo Quadro 4.2:

Quadro 4.2 – Métricas obtidas no banco de imagens com variação de iluminação e ângulo.

Método	Verdadeiro-positivos	Falso-negativos	Falso-positivos
HOG	369	217	1
MTCNN	485	101	1

Fonte: Autor.

#### 4.1.2 Testes de Reconhecimento

O banco de imagens usado para reconhecimento foi o de (HATTORI et al., 2014), no qual usa-se imagens coletadas em um ambiente similar ao do caso de uso desse trabalho, como pode-se verificar na Figura 4.1, além disso usa se imagens com 1280x960 *pixels* no formato JPEG.

O *dataset* em questão apresenta imagens em distâncias variadas, então o teste foi realizado cortando imagens com menos de 50x50 *pixels*, testou-se 15 itens, e o Quadro 4.3 apresenta os resultados onde o total de faces esperadas para serem conhecidas são 42, o teste foi realizado com distância euclidiana variação da distância euclidiana que obteve mais um desempenho já citado de mais de 99% com o valor de menor ou igual que 0,6.

Quadro 4.3 – Métricas obtidas do reconhecimento com tamanho alterado para 50x50 *pixels* no *dataset* de reconhecimento e variação da distância euclidiana máxima para reconhecimento.

Verdadeiro-positivos	Falso-negativos	Falso-positivos	Distância Euclidiana
35	1	6	0,6
39	3	0	0,5

Fonte: Autor.

Figura 4.1 – Exemplo de imagem do *dataset* utilizado com variação de iluminação e pose.



Fonte: Retirado de Hattori et al. (2014).

## 4.2 DATASET DE UMA CÂMERA DE MONITORAMENTO

As imagens existentes foram coletados de uma câmera Intelbras VIP 1120 D, seu sensor é o 6,35mm Progressive Scan CMOS sensor, e a distância focal das lentes é de 3,6 mm, com a lente sendo fixa.

As fotos foram coletadas em um corredor com a utilização do sensor de movimento da câmera e obtidas através de *softwares* genéricos na qual uma imagem tem 1280x720 *pixels* e o formato *Portable Network Graphics*.

Destaca-se que depois da coleta das imagens na câmera foram criados dois *datasets*, um que representa as maiores distâncias em relação a câmera, e se faz necessário investigar a detecção nesses pontos.

O outro *dataset* representa vários pontos em que uma pessoa apresenta a mesma distância para a câmera, variando apenas o seu ângulo com relação a câmera, as descrições dos conjuntos são melhores apresentadas no Quadro 4.4 e Quadro 4.5, uma foto do ambiente completo pode-se observar na Figura 4.2.

Quadro 4.4 – *Dataset* para distâncias maiores que seis metros, sendo a distância real considerada, o número de imagens e os pontos de amostragem.

Ponto	Distância Real	Número de imagens
1	6,1m	2
2	6,2m	2
3	6,4m	2
4	7,1m	2
5	7,1m	2
6	7,3m	2
7	8,1m	2
8	8,1m	2
9	8,3m	2

Fonte: Autor.

Sobre o Quadro 4.4 as distâncias foram calculadas usando as medições de cada ponto e utilizado a altura do indivíduo como 1,73 metros e os pontos 1,4 e 7 representam as distâncias de 6,7 e 8 metros para frente da câmera, logo com o uso do teorema de Pitágoras e sabendo as distâncias de deslocamento horizontal que é 0,9 metros entre cada ponto é possível calcular a distância real dos olhos da pessoa para a câmera.

Sobre o Quadro 4.5 a distância é sempre a mesma pois os pontos foram coletados sobre o mesmo raio, e pode-se observar os ângulos em relação a posição da câmera e a distância para a frente em relação à mesma, sendo que o ângulo aumenta do sentido horário.

Sobre o *dataset* de reconhecimento com imagens amostradas circularmente é ideal ter uma ideia do tamanho das faces que estamos analisando, o Quadro 4.6 representa o tamanho por nível de distâncias, a Figura 4.2 representa o *dataset* da amostragem circular e a Figura 4.3 representa o *dataset* com longas distâncias, a imagem total foi reduzida para facilitar o entendimento.

Quadro 4.5 – *Dataset* com amostragem circular e com mudança do ângulo entre posição da câmera e pessoa, com crescimento do ângulo sentido horário.

Ponto	Distância	Ângulo	Número de imagens
1	2m	0°	2
2	2m	11°	2
3	2m	22°	2
4	3m	0°	2
5	3m	11°	2
6	3m	22°	2
7	4m	0°	2
8	4m	11°	2
9	4m	22°	2
10	5m	0°	2
11	5m	11°	2
12	5m	22°	2

Fonte: Autor.

Quadro 4.6 – Tamanho médio das imagens a partir da detecção por distância no *dataset* amostrado circularmente.

Distância	Tamanho médio de uma face
2m	57x57px
3m	42x42px
4m	33x33px
5m	28x28px

Fonte: Autor.

Nota-se que existem variações no Quadro 4.6 nos tamanhos, devido ao fato de que existem ângulos e existe uma variação pequena na ROI encontrada, pois a face não está exatamente na mesma posição, além disso, os valores obtidos são usados para as comparações no reconhecimento.

Figura 4.2 – Representação do *dataset* de medidas circulares em diferentes distâncias, no qual a primeira marcação visível a esquerda é a de 3 metros com uma inclinação de  $0^\circ$  a câmera, e o indivíduo está a  $11^\circ$  para direita.



Fonte: Autor.

Figura 4.3 – Representação dos pontos de medidas do *dataset* de altas distâncias, indivíduo na posição mais distante em 8,3 metros.



Fonte: Autor.

### 4.2.1 Testes de detecção

Para a detecção das imagens da câmara analisou-se três casos, a configuração de tamanho mínimo para as imagens, quando usado o HOG (Quadro 4.7) a detecção consegue detectar imagens próximas de 40x40 *pixels*, pois a imagem é percorrida com uma *sliding window* de tal tamanho.

O MTCNN como também é um método que utiliza de uma pirâmide de imagens, pode-se escolher as configurações da sua *sliding window*, sendo assim foi utilizada uma de 40x40 (Quadro 4.9) e outra de 20x20 (Quadro 4.8) como tamanho mínimo a ser percorrido em uma imagem, sendo analisado primeiro o *dataset* com amostragem circular e os parâmetros dos classificadores para maiores detecções corretas.

Quadro 4.7 – Métricas obtidas do HOG com *sliding window* de 40x40 *pixels* no *dataset* amostrado circularmente.

Total de Fotos	Verdadeiro-positivos	Falso-negativos	Falso-positivos
24	19	5	0

Fonte: Autor.

Quadro 4.8 – Métricas obtidas do MTCNN com *sliding window* de 40x40 *pixels* no *dataset* amostrado circularmente.

Total de Fotos	Verdadeiro-positivos	Falso-negativos	Falso-positivos
24	24	0	2

Fonte: Autor.

Quadro 4.9 – Métricas obtidas do MTCNN com *sliding window* 20x20 *pixels* no *dataset* amostrado circularmente.

Total de Fotos	Verdadeiro-positivos	Falso-negativos	Falso-positivos
24	24	0	0

Fonte: Autor.

O valor da *sliding window* usado no HOG é de 40 *pixels* pois o treinamento da rede foi realizado para detectar imagens próximas a esse tamanho, já o MTCNN permite uma janela menor, que se faz necessária no caso de uso desse trabalho.

Nota-se que a variação da janela para um tamanho menor permite encontrar faces menores, no caso do Quadro 4.8 a janela de tamanho 40 conseguiu encontrar faces menores, mas não é sempre o caso, o Quadro 4.10 demonstra isso.

Além disso, qualquer modificação em fatores de escala do método não farão faces menores serem detectadas, pois nas escalas ocorre o *downsampling* da imagem origi-

nal, o *downsampling* gera perda da informação e esse trabalho não avalia se a perda é significativa para as detecções.

A análise para a detecção do *dataset* de imagens com distâncias maiores foi feita nos parâmetros que obtiveram melhor resultado, os valores podem ser observados no Quadro 4.10.

Quadro 4.10 – Métricas obtidas do MTCNN com *sliding window 20x20 pixels* no *dataset* de distâncias maiores que 6 metros e 6 imagens por distância.

Distâncias	Verdadeiro-positivos	Falso-negativos	Falso-positivos
6,1m - 6,4m	5	1	0
7,1m - 7,3m	4	2	0
8,1m - 8,3m	1	5	0

Fonte: Autor.

Se modificado o valor da *sliding window* para um menor é possível melhorar os resultados de detecção no *dataset* dessas distâncias, porém o trabalho relacionado de (XU; LIU; LI, 2014) menciona que em imagens com um número de *pixels* muito baixo, perde-se o poder de comparação no reconhecimento quando comparado a outras imagens, no caso o teste em questão utiliza faces detectadas menores que *20x20 pixels*, logo as imagens que são muito pequenas foram descartadas.

#### 4.2.2 Testes de reconhecimento

Nos experimentos de reconhecimento usando o *dataset* que contém amostragem circular foram testados se imagens na mesma distância e no mesmo raio reconhecem uma única pessoa (Quadro 4.11), e se comparados diferentes raios também é reconhecida uma única pessoa (Quadro 4.12), ou seja sem falsos-negativos.

Quadro 4.11 – Métricas obtidas do reconhecimento do *set* de amostragem circular, com diferentes tamanhos da imagem usando interpolação bicúbica e com distância euclidiana menor ou igual que 0,6.

Distância	Falsos-negativos	Tamanho da imagem
2m	0	57x57px
3m	0	57x57px
4m	0	57x57px
5m	0	57x57px
2m	0	28x28px
3m	0	28x28px
4m	0	28x28px
5m	0	28x28px

Fonte: Autor.

Quadro 4.12 – Métricas obtidas do reconhecimento nos conjuntos que foram amostrados circularmente, com variação de tamanho da imagem usando interpolação bicúbica e distância euclidiana menor ou igual que 0,6.

Conjuntos	Falsos-negativos	Tamanho da imagem	Distância máxima
2m,3m	0	57x57px	0,50
2m,3m	0	42x42px	0,48
2m,4m	1	57x57px	0,66
2m,4m	1	33x33px	0,63
2m,5m	1	57x57px	0,67
2m,5m	1	28x28px	0,64

Fonte: Autor.

Nota-se que foi testado um algoritmo simples de equalização de histogramas para redução dos efeitos de falta de luminosidade em fotos mais distantes ou no instante de tempo diferente, e mesmo assim o conjunto de 2m e 4m continuou com um falso negativo em uma distância de 0,61.

### 4.3 RESULTADOS OBTIDOS

Sobre a parte de Detecção, os resultados sobre as análises dos banco de dados de imagens demonstram que o método que utiliza redes neurais quando se trata de detecção tem resultados muito acima do que os que extraem recursos da imagem, é importante



notar as duas não detecções que se observam no Quadro 4.1 são de fotos extremas com uma variação do ângulo vertical da face em 90 graus.

Na questão do Quadro 4.2, a iluminação ou sua ausência fez com que as porcentagens de detecção caíam de uma maneira abrupta, para os dois métodos, o resultado corrobora de que a iluminação no mundo real é um desafio para os sistemas atuais.

Sobre os resultados dos *datasets* criados, percebeu-se que o método que utiliza redes neurais convolucionais tem um desempenho de detecção muito acima do método que não utiliza.

No conjunto criado foi possível detectar 100% das faces nas imagens nas quais existe uma visão frontal de um elemento, nota-se que o classificador não é perfeito e erros podem ocorrer quando aumentado o número de amostras como no Quadro 4.1.

O problema dos falsos-positivos pode ocorrer como ocorre no caso do Quadro 4.8, sendo possível reduzir o número de falsos-positivos se é conhecido os tamanhos que se esperam de uma face por exemplo.

Os testes de reconhecimento para um *dataset* real (Quadro 4.3) retratam a dura realidade dos exemplos de mundo sem as condições ideais, no *dataset* existem diferentes pessoas que estão conversando e os valores de reconhecimento começam a cair, dentre os três elementos que contém nas imagens de teste, somente o que não teve inclinações bruscas conseguiu ser reconhecido corretamente sempre, os outros dois estiveram envolvidos em falsos-positivos ou falsos-negativos.

Ainda no Quadro 4.3, um *threshold* de distância mais rígido (mais próximo de zero é mais rígido) ajudou a redução do número de falsos-positivos, porém aumentou o número de falsos-negativos, ocorrendo assim uma troca, as inclinações das pessoas quando analisadas em comparações um para um e em imagens com duas dimensões sendo possíveis rotações em três dimensões é uma limitação bem forte para a taxa de reconhecimento.

Os sistemas mais robustos como (CORVEE; BAK; BREMOND, 2012) utilizam de várias técnicas para a coleta do perfil de uma pessoa, o que poderia amenizar o problema dos ângulos visto que a pessoa vai ser descrita por outros fatores como a sua roupa por exemplo.

Nos testes com imagens com o *dataset* criado, pode-se observar pelos testes a capacidade de reconhecimento de um mesmo elemento quando ele está na mesma distância da câmera, o que não ocorre quando a comparação passa a ser com distâncias maiores.

Esse problema ocorre e causa uma redução das taxas de reconhecimento, segundo (XU; LIU; LI, 2014) imagens que sofrem redução do seu tamanho não representam bem uma imagem de baixa resolução obtida por uma distância maior e a capacidade de reconhecimento é reduzida rapidamente com a queda de resolução em dimensões a partir de 64x64 *pixels*.

## 5 CONCLUSÃO


Esse trabalho teve como propósito estudar técnicas de detecção e reconhecimento facial quando aplicadas sobre imagens de uma câmera de segurança do Núcleo de Ciência da Computação da Universidade Federal de Santa Maria.

Na parte do projeto de implementação foi criada uma interface parametrizada a partir dos métodos escolhidos com o intuito de uma visualização das informações obtidas a partir da utilização de dois métodos de detecção facial e a utilização de um de reconhecimento facial.

A partir da escolha dos métodos eles foram analisados com os diversos *datasets* conhecidos ou criados e se obteve um alto desempenho no que se refere à detecção facial, em comparação o reconhecimento apresentou limitações no quesito de distância das faces para a câmera e a resolução das imagens obtidas, fatores esses que fazem com que a taxa de reconhecimento desse tipo de imagem fique melhor em distâncias pequenas, como exemplo, a primeira entrada do laboratório do núcleo a partir da câmera.

Ressalta-se ainda que a aplicação pode ser utilizada para eventuais testes em imagens nas diversas etapas que existem durante um processo de detecção e reconhecimento facial ou suas características, um exemplo pode ser visualizado na Figura 5.1, na qual a interface do programa demonstra a distância entre um indivíduo sério e sorrindo.

Figura 5.1 – Exemplo de imagem da interface numa comparação de expressão facial.

Hora Detec	Foto analisada	Foto banco	Distância Max	Distância Real
9:50 p.m.			0.6	0.33

Fonte: Autor.

Como trabalhos futuros, pode-se automatizar as operações de detecção e reconhecimento de acordo com a região de melhor desempenho, visto que a consistência dos dados é preservada no programa, também é possível aplicar métodos que auxiliem em alguma etapa do processo de reconhecimento como tratamento da iluminação da mesma, ou também auxílios para os métodos de detecção.

## REFERÊNCIAS BIBLIOGRÁFICAS

AL-FEDAGHI, S.; ALRASHED, A. Schematizing uml use cases. In: **IEEE 2013 Fourth World Congress on Software Engineering**. Washington, DC, USA: IEEE Computer Society, 2013. Acesso em 10 set. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/6754260>>.

AMOS, B.; LUDWICZUK, B.; SATYANARAYANAN, M. **OpenFace: A general-purpose face recognition library with mobile applications**. [S.l.], 2016. Acesso em 29 abr. 2019. Disponível em: <<http://elijah.cs.cmu.edu/DOCS/CMU-CS-16-118.pdf>>.

BALLARD, D. H.; BROWN, C. M. **Computer Vision**. New Jersey: Prentice Hall, 1982.

BARNOUTI, N. H. Improve face recognition rate using different image pre processing techniques. **American Journal of Engineering Research(AJER)**, v. 5, n. 4, p. 46–53, 2016. Acesso em 29 abr. 2019. Disponível em: <[http://www.ajer.org/papers/v5\(04\)/E0504046053.pdf](http://www.ajer.org/papers/v5(04)/E0504046053.pdf)>.

BRAJE, W. L. et al. Illumination effects in face recognition. **Psychobiology**, v. 26, n. 4, p. 371–380, 1998. Acesso em 10 mai. 2019. Disponível em: <<https://link.springer.com/article/10.3758/BF03330623>>.

CERNA, L. R. **Fusão de descritores de histogramas de gradientes para a detecção de faces baseado em uma cascata de classificadores**. 2014. 111 f. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Ouro Preto, Ouro Preto, 2014.

CONSTANTINE, L. L.; LOCKWOOD, L. A. D. Structure and style in use cases for user interface design. In: HARMELEN, M. V. (Ed.). **Object Modeling and User Interface Design**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001. p. 245–279.

CORVEE, E.; BAK, S.; BREMOND, F. People detection and re-identification for multi surveillance cameras. **VISAPP 2012 - Proceedings of the International Conference on Computer Vision Theory and Applications**, v. 1, p. 82–88, 2012.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: 2005 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 01., 2005, San Diego. **Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01**. Washington, DC, USA: IEEE Computer Society, 2005. p. 886–893. Acesso em 20 out. 2019. Disponível em: <<https://doi.org/10.1109/CVPR.2005.177>>.

DELBIAGGIO, N. **A comparison of facial recognitions algorithms**. 2017. 45 p. Monografia (Trabalho de Conclusão de Curso) — Degree Programme in Business Information Technology, Haaga-Helia University of Applied Sciences, HAAGA-HELIA, 2017.

DINIZ, F. et al. Redface: um sistema de reconhecimento facial baseado em técnicas de análise de componentes principais e autofaces. **Revista Brasileira de Computação Aplicada**, v. 5, n. 1, p. 42–54, 2013. Acesso em 20 out. 2019. Disponível em: <<http://seer.upf.br/index.php/rbca/article/view/2627>>.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, n. 4, p. 193–202, 1980. Acesso em 27 abr. 2019. Disponível em: <<https://doi.org/10.1007/BF00344251>>.

GEORGHIADES, A.; BELHUMEUR, P.; KRIEGMAN, D. From few to many: Illumination cone models for face recognition under variable lighting and pose. **IEEE Trans. Pattern Anal. Mach. Intelligence**, v. 23, n. 6, p. 643–660, 2001. Acesso em 27 out. 2019. Disponível em: <<https://pdfs.semanticscholar.org/97bb/c2b439a79d4dc0dc7199d71ed96ad5e3fd0e.pdf>>.

GOURIER, N.; HALL, D.; CROWLEY, J. L. Estimating face orientation from robust detection of salient facial structures. In: FGNET (IST-2000-26434), 2004, Cambridge. **FG Net workshop on visual observation of deictic gestures**. Washington, DC, USA: IEEE Computer Society, 2004. v. 6, p. 7. Acesso em 20 out. 2019. Disponível em: <<http://www-prima.inrialpes.fr/perso/Gourier/Pointing04-Gourier.pdf>>.

HATTORI, K. et al. **Image Dataset for Researches about Surveillance Camera**. [S.l.], 2014. Acesso em 10 nov. 2019. Disponível em: <<http://www.consortium.ri.cmu.edu/projSRD.php>>.

HE, K. et al. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Washington, DC, USA: IEEE Computer Society, 2016. p. 770–778. Acesso em 20 out. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/7780459>>.

JAIN, V.; LEARNED-MILLER, E. **Fddb: A Benchmark for Face Detection in Unconstrained Settings**. [S.l.], 2010. Acesso em 20 abr. 2019. Disponível em: <<http://vis-www.cs.umass.edu/fddb/fddb.pdf>>.

JHINGRAN, A.; KHEDKAR, P. Analysis of recovery in a database system using a write-ahead log protocol. **ACM SIGMOD Record**, v. 21, p. 175–184, 1992.

KING, D. E. Max-margin object detection. **CoRR**, abs/1502.00046, 2015. Acesso em 20 out. 2019. Disponível em: <<http://arxiv.org/abs/1502.00046>>.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012**. Neural Information Processing Systems Foundation, Inc. (NIPS), 2012. p. 1106–1114. Acesso em 23 abr. 2019. Disponível em: <<http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>>.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. Acesso em 10 mai. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/726791>>.

LU, C.; TANG, X. Surpassing human-level face verification performance on LFW with gaussianface. **CoRR**, abs/1404.3840, 2014. Acesso em 25 abr. 2019. Disponível em: <<http://arxiv.org/abs/1404.3840>>.

PANCERI, J. et al. Reconhecimento facial baseado em hog e pca: uma comparação quanto à invariância à iluminação. **Revista Ifes Ciência-ISSN 2359-4799**, v. 1, n. 1, 2015. Acesso em 20 out. 2019. Disponível em: <<https://pdfs.semanticscholar.org/1e9d/654570e23b9edaaed6289ba159f4a4ba142c.pdf>>.

PATEL P. E.; PATIL, N. N. Testcases formation using uml activity diagram. In: **CSNT '13: Proceedings of the 2013 International Conference on Communication Systems and Network Technologies**. Washington, DC, USA: IEEE Computer Society, 2013. Acesso em 10 set. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/6524534>>.

PRADO, K. S. do. **Comparação de técnicas de reconhecimento facial para identificação de presença em um ambiente real e semicontrolado**. 2017. 172 f. Dissertação (Mestrado em Sistemas de Informação) — Universidade de São Paulo, São Paulo, 2017.

ROSENBLATT, F. **Principles of neurodynamics: perceptrons and the theory of brain mechanisms**. [S.l.]: Spartan Books, 1962.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. **CoRR**, abs/1503.03832, 2015. Acesso em 27 abr. 2019. Disponível em: <<http://arxiv.org/abs/1503.03832>>.

SILVA, A. L.; CINTRA, M. E. Reconhecimento de padrões faciais: Um estudo. In: **Encontro Nacional de Inteligência Artificial e Computacional**. Porto Alegre: SBC, 2015. p. 224–231. Acesso em 20 out. 2019. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/eniac/2015/034.pdf>>.

SUN, X.; WU, P.; HOI, S. C. H. Face detection using deep learning: An improved faster RCNN approach. **CoRR**, abs/1701.08289, 2017. Acesso em 06 mai. 2019. Disponível em: <<http://arxiv.org/abs/1701.08289>>.

SUN, Y. et al. Deep learning face representation by joint identification-verification. In: **Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2**. Cambridge, MA, USA: MIT Press, 2014. p. 1988–1996. Acesso em 24 abr. 2019. Disponível em: <<https://arxiv.org/abs/1406.4773>>.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. London: Springer, 2011.

VASCONCELOS, C. N.; CLUA, E. W. G. Deep learning - teoria e prática. In: Sociedade Brasileira de Computação - SBC (Ed.). **Jornadas de Atualização em Informática 2017**. 1. ed. Porto Alegre/RS: Sociedade Brasileira de Computação - SBC, 2017. cap. 6, p. 212–260. ISBN 978-85-7669-374-1. Acesso em 25 abr. 2019. Disponível em: <<http://csbc2017.mackenzie.br/public/files/all/livro-jai.pdf>>.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. **Comput. Vis. Pattern Recog**, v. 1, 2001. Acesso em 24 abr. 2019. Disponível em: <<https://ieeexplore.ieee.org/document/990517>>.

WANG, M.; DENG, W. Deep face recognition: A survey. **CoRR**, abs/1804.06655, 2018. Acesso em 20 out. 2019. Disponível em: <<http://arxiv.org/abs/1804.06655>>.

WOZNIAK, M.; GRAÑA, M.; CORCHADO, E. A survey of multiple classifier systems as hybrid systems. **Information Fusion**, v. 16, p. 3–17, 2014.

XU, X.; LIU, W.; LI, L. Low resolution face recognition in surveillance systems. **Journal of Computer and Communications**, v. 02, p. 70–77, 2014. Acesso em 10 mai. 2019. Disponível em: <<https://www.scirp.org/journal/paperinformation.aspx?paperid=41873>>.

YANG, S. et al. WIDER FACE: A face detection benchmark. **CoRR**, abs/1511.06523, 2015. Acesso em 23 abr. 2019. Disponível em: <<http://arxiv.org/abs/1511.06523>>.

ZHANG, C.; ZHANG, Z. **A Survey of Recent Advances in Face Detection**. [S.l.], 2010. Acesso em 20 out. 2019. Disponível em: <<https://www.microsoft.com/en-us/research/publication/a-survey-of-recent-advances-in-face-detection/>>.

ZHANG, F. et al. Accurate face detection for high performance. **CoRR**, abs/1905.01585, 2019. Disponível em: <<http://arxiv.org/abs/1905.01585>>.

ZHANG, K. et al. Joint face detection and alignment using multitask cascaded convolutional networks. **IEEE Signal Processing Letters**, v. 23, n. 10, p. 1499–1503, 2016. Acesso em 23 abr. 2019. Disponível em: <<https://arxiv.org/pdf/1604.02878.pdf>>.

ZHANG, S. et al. S<sup>3</sup>fd: Single shot scale-invariant face detector. **CoRR**, abs/1708.05237, 2017. Acesso em 25 abr. 2019. Disponível em: <<http://arxiv.org/abs/1708.05237>>.

## ANEXO A – EXEMPLOS E LOCAIS DOS SETS

O *set* de imagens construído circularmente pode se observar na Figura A.1 e no Github<sup>1</sup>, e o *set* de distâncias maiores na Figura A.2 e no Github<sup>2</sup>.

Figura A.1 – Exemplo de fotos do *dataset* amostrado circularmente em 2 metros de distância da câmera.



Fonte: Autor.

<sup>1</sup>[github.com/ftfajardo/Dataset1](https://github.com/ftfajardo/Dataset1)

<sup>2</sup>[github.com/ftfajardo/Dataset2](https://github.com/ftfajardo/Dataset2)



Figura A.2 – Exemplo de fotos do *dataset* de distâncias maiores com exemplo de 6 e 7 metros de distância para a câmera.



Fonte: Autor.