

Érica Kido Shimomoto

**Sistema de Reconhecimento Facial com
abordagem *open-set* aplicado ao controle de
acesso**

Manaus - Amazonas

Novembro, 2014

Érica Kido Shimomoto

**Sistema de Reconhecimento Facial com abordagem
open-set aplicado ao controle de acesso**

Monografia submetida à banca avaliadora do Curso de Engenharia de Controle e Automação, como pré-requisito para a obtenção do título de Engenheiro em Controle e Automação pela Universidade do Estado do Amazonas, Escola Superior de Tecnologia.

Universidade do Estado do Amazonas - UEA
Escola Superior de Tecnologia
Programa de Graduação em Engenharia de Controle e Automação

Orientador Almir Kimura Junior, Msc.

Manaus - Amazonas
Novembro, 2014

Érica Kido Shimomoto

Sistema de Reconhecimento Facial com abordagem *open-set* aplicado ao controle de acesso

Monografia submetida à banca avaliadora do Curso de Engenharia de Controle e Automação, como pré-requisito para a obtenção do título de Engenheiro em Controle e Automação pela Universidade do Estado do Amazonas, Escola Superior de Tecnologia.

Trabalho aprovado. Manaus - Amazonas, 29 de novembro de 2014:

Almir Kimura Junior, Msc.
Orientador

Professor
Dr. Walter A. Vermehren Valenzuela

Professor
Dr. Daniel Guzmán del Rio

Professor
MSc. Marcelo Albuquerque de Oliveira

Manaus - Amazonas
Novembro, 2014

Agradecimentos

A minha família e amigos pela paciência e apoio durante o processo de produção desta monografia. Ao meu orientador, professor MSc. Almir Kimura Junior, por ter abraçado a minha ideia e me guiado na elaboração e redação deste trabalho. Ao nosso coordenador, professor Dr. Walter Valenzuela, que apesar das duras palavras, sempre tentou me mostrar o melhor caminho. À Universidade do Estado do Amazonas e ao seu corpo docente, que dedicou seu tempo e esforço para me passar os ensinamentos que me tornaram a profissional que sou hoje. Em especial, ao meu supervisor de estágio, MSc. Ruan Belém, que me engajou no projeto que deu origem a esta monografia e a toda equipe do RD2 da Envision que me auxiliou com testes e me proveram de valiosas opiniões e críticas que ajudaram este trabalho a crescer.

Resumo

Apesar do grande enfoque que o reconhecimento facial tem recebido no mundo da computação visual, grande parte das técnicas desenvolvidas para solucionar esta questão encaram o reconhecimento facial como um problema de *closed-set*, onde apenas amostras de teste pertencentes a base de dados de treinamento são consideradas. Entretanto, para aplicações reais, o reconhecimento facial é um problema de *open-set*, onde amostras de pessoas não pertencentes a base de dados podem ser testadas pelo reconhecedor. Este trabalho tem como objetivo apresentar um novo meio de se solucionar este problema utilizando a Regressão por Vetores de Suporte. Além disso, foi proposto um novo método de detecção facial que combina algumas técnicas de detecção de objetos, com o objetivo de reduzir o tempo de detecção quando comparado ao método puro de Viola-Jones, dentro do contexto do sistema. Por fim, a implementação de um *software* de reconhecimento facial, ligado através da tecnologia sem fio *ZigBee* a um circuito de acionamento de uma fechadura elétrica, simulando o controle de acesso de uma sala.

Palavras-chaves: Reconhecimento Facial. Aprendizado de Máquinas. Detecção Facial. Regressão por Vetores de Suporte.

Abstract

Despite the great attention that the Facial Recognition problem has received in the world of visual computing, the biggest part of the techniques developed to solve this issue see the facial recognition as a closed-set problem, where it considers only test samples from people registered in the database. However, in real world applications, facial recognition is an open-set problem, where samples from people not registered in the database can be tested by the recognizer. This work presents a new way to solve open-set recognition problems by using the Support Vector Regression. Besides, it proposes a new facial detection method, which combines some object detection techniques in order to reduce the detection time when compared with the pure Viola-Jones method, under the system environment. Lastly, the implementation of a facial recognition software, connected using *ZigBee* to a driving circuit of an electric lock, simulating a room's access control.

Key-words: Facial Recognition. Machine Learning. Facial Detection. Support Vector Regression.

Lista de ilustrações

Figure 1 – Passos no PDI	16
Figure 2 – Componentes de um sistema de PDI	17
Figure 3 – Haar <i>features</i>	21
Figure 4 – <i>features</i> em um rosto	22
Figure 5 – LBP para imagens com mudanças monotônicas de cinza.	25
Figure 6 – Padrão Binário Local.	26
Figure 7 – Padrões encontrados pelo LBP.	27
Figure 8 – Vizinhança com diferentes tamanhos de R.	27
Figure 9 – Separação dos dados pelo SVM. a) Dados não linearmente separáveis. b) Ao jogar esses dados para um espaço de maior dimensionalidade, é possível separar estes dados por meio de um hiperplano.	28
Figura 10 – Exemplo de Regressão por Vetores de Suporte.	30
Figura 11 – Matriz Confusão.	31
Figura 12 – Exemplo de curva ROC. A) Curva Ideal. B) Bom Teste. C) Curva randômica.	32
Figura 13 – Fluxograma A - Aprendizado.	37
Figura 14 – Fluxograma B - Reconhecimento.	38
Figura 15 – Circuito de Acionamento da fechadura.	40
Figura 16 – Ligação da fechadura elétrica com o circuito de acionamento e o microcontrolador.	40
Figura 17 – Fluxograma 1 - Detector de Rostos.	41
Figura 18 – Histograma das cores de pele.	44
Figura 19 – Histograma das cores que não são de pele.	44
Figura 20 – Histograma das cores em geral.	45
Figura 21 – Segmentação da pele.	46
Figura 22 – Rosto Detectado.	47
Figura 23 – Fluxograma 2 - Processamento do Rosto.	49
Figura 24 – Encontrando o vetor que tem a mesma direção que a reta que liga os dois olhos.	50
Figura 25 – Cálculo do ângulo theta.	50
Figura 26 – Imagem rotacionada.	51
Figura 27 – Imagem recortada.	51
Figura 28 – Fluxograma 2.2 - Normalização da Iluminação.	52
Figura 29 – Correção Gamma.	53
Figura 30 – Filtro Diferença de Gaussiana.	54

Figura 31 – Contrast Equalization.	55
Figura 32 – Fluxograma 3.a - Treinamento do Reconhecedor.	59
Figura 33 – Fluxograma 3.b - Reconhecimento.	60
Figura 34 – <i>Shape Model</i> - Subespaço representado pelo modelo linear.	62
Figura 35 – Rastreador de Rosto.	62
Figura 36 – Curva ROC para o Segmentador de Pele.	63
Figura 37 – Segmentador de Pele - Ocorrência de falso positivo.	64
Figura 38 – Segmentador de Pele - Ocorrência de falso negativo.	65
Figura 39 – Segmentador de Pele - Imagens corretamente segmentadas.	65
Figure 40 – Detecção pelo método proposto - a) Verdadeiro Positivo e b)Falso Positivo	69
Figure 41 – Detecção pelo método proposto - Detecção em rostos com óculos. . . .	69
Figura 42 – Acurácia do SVR de acordo com o número e tamanho das amostras. . .	71
Figura 43 – Tempo de Treinamento do SVR de acordo com o número e tamanho das amostras.	72
Figura 44 – Especificidade do SVR de acordo com o número e tamanho de amostras	73
Figura 45 – Imagens em nível de cinza usadas no treinamento do SVR.	74
Figura 46 – Imagens com iluminação normalizada usadas no treinamento do SVR. .	75
Figura 47 – Imagens com LPB usadas no treinamento do SVR.	76
Figura 48 – Imagens com iluminação normalizada e LBP usadas no treinamento do SVR.	77
Figura 49 – Imagens com filtro Gamma e LBP usadas no treinamento do SVR. . .	78
Figura 50 – Gráfico comparativo da acurácia do reconhecedor treinado com dife- rentes características.	79
Figura 51 – Megumi - Tela Principal.	82
Figura 52 – Megumi - Carregar vídeo de um arquivo.	83
Figura 53 – Megumi - Reconhecimento.	83
Figura 55 – Megumi - Tela com acesso de administrador.	84
Figura 54 – Megumi - Tela de <i>log in</i>	84
Figura 57 – Megumi - Obtendo amostras do novo perfil.	85
Figura 56 – Megumi - Adicionar novo perfil.	85
Figura 58 – Megumi - Deletando um perfil.	86
Figura 59 – Computador ligado ao <i>ZigBee</i> Coordenador.	87
Figura 60 – <i>ZigBee</i> Coordenador.	87
Figura 61 – Controle da fechadura.	88
Figura 62 – Microcontrolador ligado ao ZED e ao circuito de acionamento da fe- chadura.	88
Figura 63 – Parte física completa.	89

Lista de tabelas

Tabela 1 – Valores retornados pelo SVR.	58
Tabela 2 – Segmentador de Pele ($\alpha=0,4$).	64
Tabela 3 – Detector de Rosto - Apenas com classificadores Haar em Cascata.	66
Tabela 4 – Detector de Rosto - Método proposto.	66
Tabela 5 – Tempos (ms) de detecção pelo Classificador Haar em Cascata	67
Tabela 6 – Tempos (ms) de detecção pelo método proposto	68
Tabela 7 – Reconhecedor de Rosto - Utilizando Nível de Cinza.	74
Tabela 8 – Reconhecedor de Rosto - Utilizando Nível de Cinza da Imagem normalizada.	75
Tabela 9 – Reconhecedor de Rosto - Utilizando Padrões Binários Locais.	76
Tabela 10 – Reconhecedor de Rosto - Utilizando Padrões Binários Locais e a Normalização da Iluminação.	77
Tabela 11 – Reconhecedor de Rosto - Utilizando filtro Gamma e LBP.	78
Tabela 12 – Desempenho do reconhecedor de acordo com o número de pessoas registradas - Um Contra Todos convencional.	79
Tabela 13 – Desempenho do reconhecedor de acordo com o número de pessoas registradas - Um Contra Todos balanceado.	80
Tabela 14 – Tempo de Treinamento para as duas formas da abordagem Um Contra Todos em (s).	80
Tabela 15 – Número médio de quadros analisados para tomar uma decisão.	81
Tabela 16 – Medidas de desempenho para o reconhecedor.	81

Sumário

	Lista de ilustrações	6
	Lista de tabelas	8
	Sumário	9
	Introdução	11
1	REFERENCIAL TEÓRICO	15
1.1	Processamento Digital de Imagem	15
1.1.1	Principais Passos no Processamento Digital de Imagem	15
1.1.2	Componentes de um sistema de processamento de imagens	17
1.2	Espaço de cores	18
1.3	Detecção Facial	20
1.3.1	Classificadores Haar em Cascata	21
1.4	Reconhecimento Facial	23
1.4.1	Extratores de Características	24
1.4.1.1	Padrões Binários Locais	25
1.4.2	Máquinas de Vetores de Suporte	27
1.4.2.1	Classificação por Vetores de Suporte	28
1.4.2.2	Regressão por Vetores de Suporte	29
1.5	Medidas Básicas de Avaliação de Performance de Classificadores	31
1.6	Hardware	32
1.6.1	Microcontroladores	32
1.6.1.1	Processador	33
1.6.1.2	Memórias	33
1.6.1.3	Unidade de entrada e saída	33
1.6.2	Tecnologia <i>ZigBee</i>	34
1.6.2.1	Tipos de dispositivos <i>ZigBee</i>	34
2	METODOLOGIA	36
2.1	Sistema de Reconhecimento Facial - Funcionamento Geral	36
2.1.1	<i>Software</i>	36
2.1.1.1	Aprendizado	37
2.1.1.2	Reconhecimento	38
2.1.2	<i>Hardware</i>	39
2.2	Detector de Rostos	41

2.2.1	Segmentação da Pele	43
2.2.2	Deteção dos Elementos do Rosto	46
2.3	Processamento do Rosto	48
2.3.1	Normalização da Rotação	49
2.3.2	Normalização da Iluminação	52
2.3.2.1	Correção Gamma	52
2.3.2.2	Filtro Diferença de Gaussiana	53
2.3.2.3	Equalização de Contraste	54
2.3.3	Normalização da Escala	55
2.4	Reconhecedor de Rostos	55
2.4.1	Rastreador de rostos	61
3	RESULTADOS OBTIDOS	63
3.1	Segmentador de Pele	63
3.2	Detector de Rosto	65
3.3	Reconhecedor de Rosto	70
3.3.1	Tamanho e número de amostras para treinamento	70
3.3.2	<i>Features</i>	73
3.3.3	Número de pessoas registradas	79
3.3.4	Predição a partir de vários quadros	81
3.4	Sistema Implementado	82
3.4.1	<i>Software</i>	82
3.4.2	<i>Hardware</i>	86
4	CONSIDERAÇÕES FINAIS	90
4.1	Trabalhos Futuros	91
	Referências	92

Introdução

O Reconhecimento automático facial em imagens é um dos tópicos mais estudados na área da visão computacional (ZHANG; ZHANG, 2010) por ser útil em aplicações de identificação de pessoas, vigilância, ambientes inteligentes, entre outros. Devido à extensa pesquisa na área, já é possível conseguir altas taxas de acertos em reconhecimento facial em ambientes controlados (PHILLIPS et al., 2005).

Problemas de identificação em *closed-set*, ou seja, deduzir a identidade de uma imagem de teste a partir de um conjunto de identidades conhecidas, onde se sabe que esta amostra pertence a uma das identidades dentro da base dados tem sido extensamente estudados. Entretanto, para os problemas de identificação em *open-set*, onde se deve conseguir determinar se a amostra de teste é pertencente a uma pessoa conhecida ou não, pouco estudo foi realizado.

A aplicação de sistemas de reconhecimento facial no controle acesso representa um dos exemplos mais clássicos desse tipo de situação. Apesar de parecer um problema de classificação binária, onde a classe positiva corresponde a um indivíduo conhecido e a negativa, a todos os outros indivíduos desconhecidos, isso não é verdade pois não se tem amostras suficientes para modelar o desconhecido. Mesmo que alguém juntasse amostras negativas que cobrissem toda a população mundial, isso seria inviável devido ao grande desbalanceamento da base de dados de treinamento.

Formulação do Problema

O grande desafio de reconhecer o desconhecido está na dificuldade da sua modelagem. Apesar de pouco estudo ter sido feito em cima dos problemas de reconhecimento em *open-set*, alguns trabalhos, como o de (TAX; DUIN, 2004), apresentaram taxas de até 99% de reconhecimento em uma base de dados com informações da ionosfera, ao criar um método que procura encontrar uma boa descrição de uma classe conhecida, de forma a ser possível a identificação das classes desconhecidas. (LI; WECHSLER, 2005) utilizou uma abordagem baseada na transdução, rejeitando uma amostra de teste de acordo com os valores de distribuição de credibilidade obtidos de um modelo de *k-nearest neighbors*. (STALLKAMP; EKENEL; STIEFELHAGEN, 2007) utilizou pontos de confiança acumulados que são utilizados para realizar o reconhecimento do desconhecido. Mais recentemente, (SCHEIRER et al., 2013; SCHEIRER; JAIN; BOULT, 2014) realizou uma formulação de problemas em *open-set*, desenvolvendo modelos probabilísticos para o reconhecimento de amostras desconhecidas. Verificou-se que estes autores utilizam em sua maioria as Máquinas de Vetores de Suporte como método de classificação.

A Regressão por Vetores de Suporte é amplamente utilizada na modelagem de séries de tempo como, por exemplo, a maré dos oceanos e o valor de fechamento de bolsas de valores. De acordo com a formulação de problemas de regressão, é possível encontrar uma função que consiga aproximar todos os pares (x,y) de onde se deseja extrair uma função (FARAG; MOHAMED, 2004). Tendo isso em mente, vê-se a possibilidade da aplicação do SVR para previsão da identidade de um rosto, com a criação de uma função de rostos. Entretanto, até onde a pesquisa bibliográfica deste trabalho abrangiu, não se encontrou um trabalho que utilizasse esta abordagem.

Justificativa

O controle de acesso a locais restritos pode ser feito de diversas formas. Dentre as mais comuns, cita-se a utilização de chaves e cartões magnético. Apesar de serem eficientes no sentido de que somente quem possui a chave ou o cartão magnético consegue acessar o local, todos estes métodos são passíveis de perda, roubo ou podem ser repassados para outras pessoas não autorizadas. Ou seja, qualquer um que possua estas ferramentas consegue o acesso.

Uma forma de se solucionar este problema é utilizar a biometria das impressões digitais como meio de autenticação para o acesso. Teoricamente, apenas a pessoa com sua biometria registrada tem acesso ao local, não existindo a possibilidade de alguém não registrado conseguir este acesso, pois ela é única para cada pessoa e não tem como ser perdida, roubada ou emprestada. Entretanto, existem alguns casos de fraude, com a utilização de próteses de silicone (HUTCHINSON, 2013).

O estudo do reconhecimento facial se justifica pois, assim como a impressão digital, não é passível de perda, roubo ou empréstimo, uma vez que o rosto é também uma característica única de cada indivíduo (com exceção de gêmeos univitelinos). Ele também se mostra mais interessante por ser um método não invasivo, que não necessita de contato com a pessoa para realizar a verificação, reduzindo riscos de contaminação e proliferação de bactérias.

Motivação

O reconhecimento facial é, se bem executado, um dos métodos de controle de acesso mais seguros pois não depende do uso de objetos de identificação, sendo imune a perdas, esquecimento ou uso indevido (alguém não autorizado utilizando a chave de uma pessoa autorizada). Além disso, ele é resistente a fraudes, uma vez que cada pessoa só possui um rosto e a construção de um modelo réplica do mesmo é bem mais complexa que apenas clonar um cartão magnético.

Entretanto, para que este reconhecimento seja devidamente realizado, é preciso tratá-lo como um problema de *open-set*. Sua maior dificuldade se encontra na identificação de uma amostra como não pertencente à base de treinamento. Devido a isto, muitos dos problemas de reconhecimento, inclusive o facial, têm sido modelados como problemas de *closed-set*. Sua abordagem como um problema de *open-set* foi pouco explorada, apesar da sua resolução ser fundamental para um efetivo reconhecimento em condições reais, onde o reconhecedor deve rejeitar amostras não pertencentes a pessoas registradas na base de dados. O desenvolvimento de uma técnica de reconhecimento em *open-set* se mostra, portanto, de grande importância para o avanço da visão computacional.

Outro ponto muito importante a se ressaltar é o fato de, entre os poucos estudos realizados para este tipo de problema, nenhum deles, até onde a pesquisa bibliográfica deste trabalho abrangeu, utilizou SVR como principal ferramenta para o reconhecimento, sendo este trabalho uma inovação no modo de se solucionar o reconhecimento facial.

Objetivos

Implementar um sistema de reconhecimento facial para o controle de acesso de uma sala que além de identificar corretamente as pessoas registradas na base de dados, consiga reconhecer quando uma pessoa não faz parte da mesma, utilizando a Regressão por Vetores de Suporte. De forma mais específica, deve-se:

- Implementar um detector facial rápido o suficiente para utilização em aplicações reais;
- Implementar um reconhecedor facial utilizando SVR;
- Criar um software utilizando a linguagem C++;
- Conectar este sistema via Zigbee a um circuito que controla a abertura e fechamento de uma fechadura elétrica, simulando o controle de acesso a uma sala;
- Realizar testes e verificar o desempenho do reconhecedor obtido.

Organização do Trabalho

Este trabalho está dividido em quatro capítulos. Inicialmente, ele aborda uma breve introdução sobre os problemas de reconhecimento facial em *open-set* e conceitua o projeto por meio da justificativa, motivação e descrição do problema. Também traz os objetivos a serem alcançados.

No Capítulo I se dá a apresentação do referencial teórico necessário ao desenvolvimento do projeto. Também é dada uma contextualização sobre os principais trabalhos

relacionados à detecção e reconhecimento facial, explicando de forma mais detalhada os pontos a serem aplicados no projeto.

Em seguida, o Capítulo II descreve o processo de desenvolvimento do sistema de reconhecimento facial. Uma visão geral sobre o sistema é dada, com uma explicação sobre o desenvolvimento de cada uma das componentes do mesmo, como a detecção e reconhecimento facial, além da implementação da parte física.

O Capítulo III apresenta os resultados obtidos, dos quais se faz uma análise e avaliação da performance dos métodos propostos.

Por último, o Capítulo IV apresenta as conclusões feitas deste trabalho, assim como propostas para trabalhos futuros.

1 Referencial Teórico

Uma vez que o principal objetivo deste projeto é a implementação de um sistema de reconhecimento facial, baseado no processamento digital de imagem, é necessário entender os principais passos para esta tarefa, além dos principais componentes que compõem este tipo de sistema.

Além disso, é preciso conhecer as técnicas e métodos específicos necessários para que o reconhecimento facial seja realizado. Falar-se-á sobre as principais técnicas de detecção e reconhecimento facial, além da aplicação de algumas técnicas de inteligência artificial necessárias.

É preciso também conhecer alguns conceitos para avaliação de desempenho dos classificadores e reconhecedores, de modo a, posteriormente, avaliar os resultados obtidos no trabalho.

Por fim, é necessário conhecer um pouco sobre o microcontrolador e tecnologia sem fio *ZigBee*, a serem utilizados na implementação física do sistema.

1.1 Processamento Digital de Imagem

Processamento Digital de Imagem (PDI) é a manipulação de uma imagem por um computador, onde a entrada é uma imagem e a saída pode ser uma imagem ou atributos extraídos da mesma. Uma imagem pode ser definida com uma função de duas dimensões $f(x,y)$, onde x e y são as coordenadas espaciais em um plano e a amplitude de $f(x,y)$ é a intensidade ou nível de cinza naquele ponto (GONZALES; WOODS, 2002). Se os valores de $f(x,y)$ forem finitos e discretos, pode-se chamar a imagem de imagem digital.

O sistema visual humano possui uma notável capacidade de reconhecer padrões. Contudo, ele dificilmente é capaz de processar o enorme volume de informação presente numa imagem. Vários tipos de degradações e distorções, inerentes aos processos de aquisição, transmissão e visualização de imagens, contribuem para limitar ainda mais essa capacidade do olho humano.

Desta forma, o principal objetivo do PDI é remover estas barreiras, de modo a facilitar a extração de informações a partir de imagens (QUEIROZ, 2003).

1.1.1 Principais Passos no Processamento Digital de Imagem

De acordo com (GONZALES; WOODS, 2002), os processos envolvidos pelo processamento digital de imagem podem ser sumarizados de acordo com o seguinte diagrama:

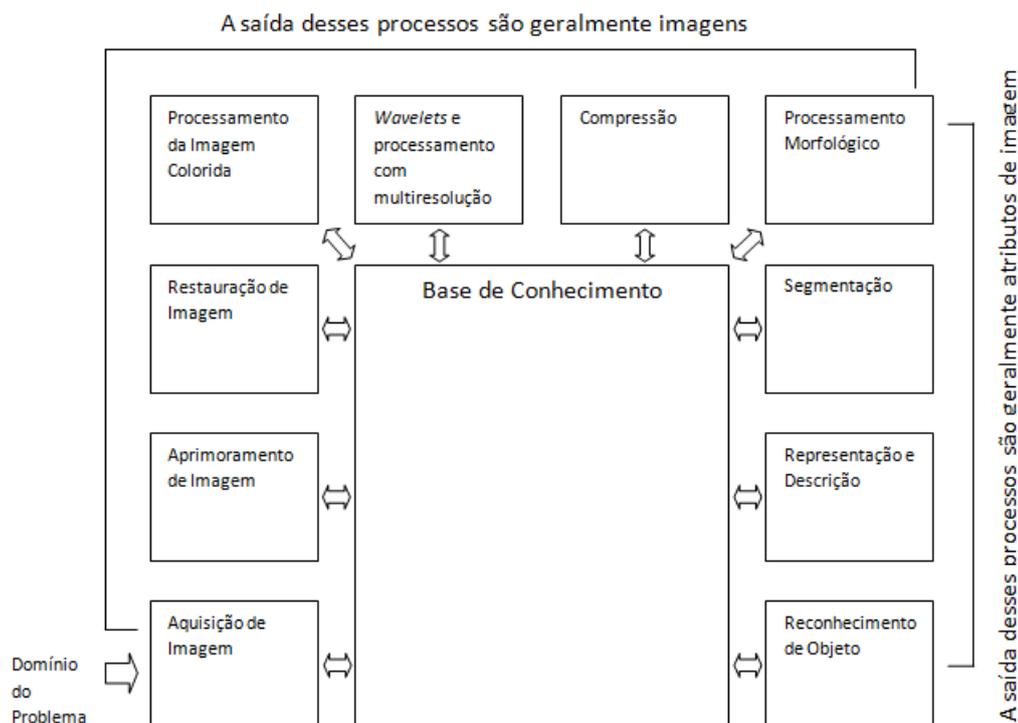


Figura 1 – Passos no PDI

Fonte: (GONZALES; WOODS, 2002)

O primeiro processo a ser realizado é a **Aquisição de Imagem**. Essa aquisição pode ser feita desde através do uso de algum artifício para capturá-la (como uma câmera) como apenas acessar uma imagem já digitalizada. Normalmente, esse processo envolve um pré processamento da imagem, como o redimensionamento dela.

O **Aprimoramento de Imagem** tem como objetivo destacar detalhes que estejam obscurecidos na imagem ou apenas ressaltar detalhes que sejam de interesse para a aplicação como, por exemplo, ressaltar as bordas de uma imagem.

A **Restauração de Imagem** é bem similar ao aprimoramento, entretanto, a melhora da aparência da imagem não está sujeita a um determinado interesse, baseado em modelos matemáticos ou probabilísticos de degradação de imagem.

O **Processamento de Imagem Colorida** utiliza as informações das cores dos pixels, sendo muito importante para a extração de características de uma imagem. Existem diversos modelos de cores nos quais as imagens podem ser representadas, sendo o mais conhecido o modelo *Red, Green, Blue* (RGB). Este processo é de grande importância para o projeto, uma vez que parte da detecção facial a ser realizada pelo sistema é baseada na cor da pele. Mais detalhes sobre isso serão discutidos mais à frente.

Wavelets são a base para a representação da imagem em vários graus de resolução.

Compressão lida com as técnicas para redução do espaço necessário para se salvar

uma imagem ou a largura de banda necessária para a transmissão da mesma.

Processamento Morfológico lida com ferramentas para a extração de componentes que consigam representar e descrever uma forma.

Segmentação é o procedimento de divisão de uma imagem em partes ou objetos que a constituem. Este é um processo difícil de ser automatizado, mas que se bem realizado aumenta as chances de, por exemplo, o reconhecimento de um objeto ser bem sucedido.

Representação e Descrição quase sempre seguem a segmentação. A segmentação retorna apenas valores de pixels. Esses pixels podem representar o contorno de uma forma ou uma forma como um todo. A descrição ou extração de atributos lida com a extração de qualidades que representem alguma informação quantitativa sobre o objeto a ser analisado.

Reconhecimento é o processo que designa uma etiqueta ou *label* a um objeto de acordo com seus descritores.

Por fim, a **base de conhecimento** agrega os conhecimentos sobre o problema.

1.1.2 Componentes de um sistema de processamento de imagens

(GONZALES; WOODS, 2002) também explica sobre os componentes de um sistema de propósito geral usado para o processamento digital de imagens, que estão representados na figura abaixo:

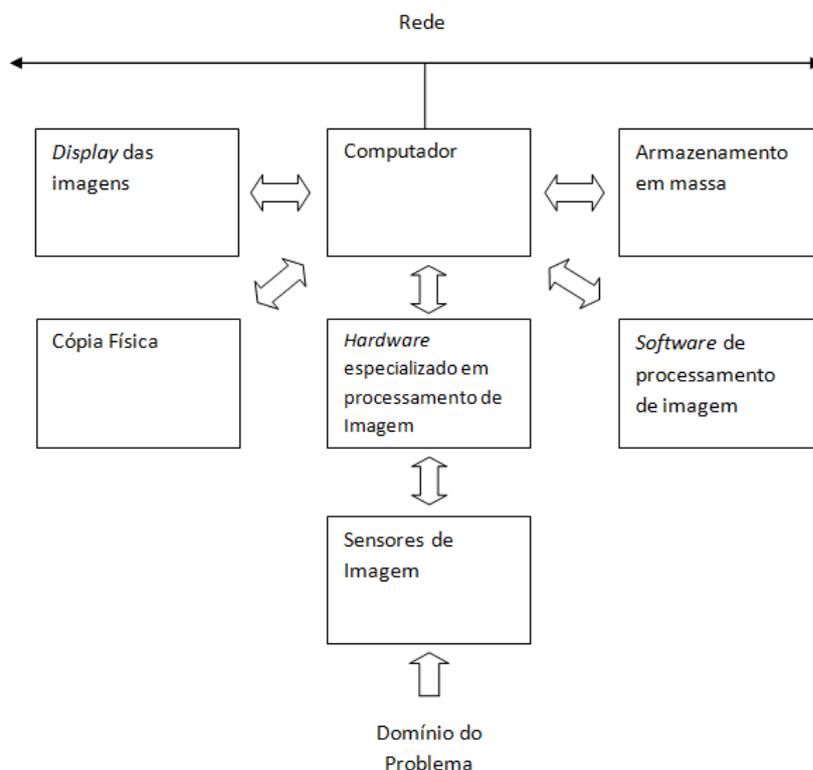


Figura 2 – Componentes de um sistema de PDI

Fonte: (GONZALES; WOODS, 2002)

Os **sensores de imagem** envolvem dois elementos: Um dispositivo físico e um digitalizador. O dispositivo físico deve ser sensível à energia emanada pelo objeto do qual se quer obter uma imagem. O digitalizador converte a saída do dispositivo físico para a forma digital.

O **hardware especializado em processamento de imagem** normalmente consiste do digitalizador, assim como no sensor, mais um *hardware* que executa operações primitivas, tais como uma ULA (Unidade Lógica-Aritmética). A ULA pode ser utilizada para a redução de ruídos na imagem a ser processada. Este *hardware* também é chamado de *front-end subsystem* e sua principal característica é a velocidade.

O **computador** é um computador de propósito geral que pode ser desde um computador pessoal a um super computador. Em algumas aplicações, é necessário um computador especialmente designados para o processamento de imagem, de modo a se alcançar um determinado nível de performance.

O **software** consiste de diversos módulos que trabalham juntos para fazer alguma tarefa.

O **armazenamento em massa** é essencial em um sistema de processamento de imagem pois, conforme os processos vão acontecendo, é possível que milhares de imagens secundárias sejam geradas e o armazenamento delas pode se tornar crítico, caso não se tenha espaço em disco suficiente. Este armazenamento se divide em três tipos: (1) Curto tempo, durante o processamento; (2) *Online*, para um rápido *recall*; (3) Armazenamento de arquivos, caracterizado pelo acesso não muito frequente.

Display de imagens em geral são monitores televisores coloridos, em geral, tela plana, onde as saídas de imagem e gráficos são mostrados.

Dispositivos de cópia física incluem impressoras, câmeras de filme, dispositivos sensíveis ao calor e unidades digitais, como CR-ROMs.

Por fim, a **rede** é responsável pela comunicação entre sistemas (outros computadores, outros processos). Devido à grande quantidade de dados em um processamento de imagem, o principal aspecto que deve ser observado na transmissão de imagens é a largura de banda. Em redes dedicadas, isso normalmente não é um problema, mas a comunicação com sites remotos via *Internet* nem sempre é eficiente.

1.2 Espaço de cores

A dificuldade em atribuir números a diferentes reações do cérebro a um estímulo visual torna a definição de cor algo difícil, muito subjetivo. Os espaços de cores têm como objetivo ajudar nesse processo de definição e descrição das cores.

É possível descrever a cor medindo sua distribuição do poder espectral (a intensidade da radiação eletro-magnética visível em vários comprimentos de onda discretos), mas em geral isso leva a um grande grau de redundância. Isso acontece porque a retina do olho faz

uma amostragem das cores usando apenas três bandas, correspondentes ao vermelho, verde e azul claro (FORD; ROBERTS, 2014). O sinal dessas células sensíveis a cor, chamadas cones, junto com o sinal dos bastonetes, sensíveis a intensidade, são combinados no cérebro para dar diversas sensações de cor. Essas sensações foram definidas da seguinte forma, de acordo com (HUNT; POINTER; POINTER, 2011):

- Brilho (*Brightness*): Indica que área exibe mais ou menos luz;
- Matiz (*Hue*): Indica as áreas que parecem ser mais similares a uma ou à proporção de duas das cores vermelho, amarelo, verde e azul;
- Coloração (*Colourfulness*): Indica qual área tem mais ou menos da sua matiz;
- Luminosidade (*Lightness*): Indica o brilho de uma imagem, em relação a um branco no cenário;
- Croma (*Chroma*): Coloração de uma área em relação ao brilho de um branco de referência;
- Saturação (*Saturation*): A coloração de uma área, relativo ao seu brilho.

A teoria das três cores (SMELSER; BALTES et al., 2001) descreve como três luzes diferentes, a vermelha, a verde e a azul, podem se combinar para formar qualquer cor visível, baseado na utilização do olho de três sensores de cor. Por este motivo, na fotografia e impressão, são usadas três cores bases e também o motivo para a maioria dos espaços de cores utilizados pelos computadores terem três parâmetros para definir uma cor.

Um espaço de cor é um método onde podemos especificar, criar e visualizar as cores, através da definição dos valores desses três parâmetros, que indicam a posição desta cor no espaço de cores. Eles não nos informam diretamente qual é a cor, pois isso depende do espaço de cor que está sendo utilizado.

Um dos espaços mais famosos e mais utilizados pelos computadores é o RGB (*Red, Green, Blue*). Cada uma dessas letras representa um dos parâmetros desse espaço de cor, sendo eles a intensidade de vermelho, verde e azul. Os valores para cada um desses parâmetros variam de 0 a 255, ou seja, 8 bits por cor, dando uma profundidade de 24 bits para cada cor. Seu espaço complementar é o CMY, formando pelo ciano ($C = 255 - R$), magenta ($M = 255 - G$) e amarelo ($Y = 255 - B$). O espaço RGB é utilizado para mostrar imagens coloridas nas telas de um computador, enquanto que o CMY é utilizado em impressoras.

Outro espaço de cor muito utilizado na computação gráfica é o HSV (*Hue, Saturation, Value*). Seus componentes representam a matiz, saturação e brilho.

Para os propósitos deste trabalho, é importante conhecer o espaço de cores YCrCb. Onde Y indica o componente Luma, ou seja, o nível de luminância, Cr é o componente

diferença vermelho e Cb é o componente diferença azul (FEIPENG, 2011). A vantagem em utilizar este espaço de cores está no fato de que se ignorarmos o componente Luma, conseguimos uma maior robustez a mudanças de iluminação em uma imagem.

1.3 Detecção Facial

É importante frisar a diferença de dois termos, erroneamente utilizados para designar o mesmo processo. A detecção facial é o processo de identificar faces, sem atribuir uma identidade a elas. Já o reconhecimento facial é o processo de identificar as faces, atribuindo-lhes suas identidades. São termos distintos mas que se completam, uma vez que para que seja possível o reconhecimento facial, é preciso primeiramente executar a detecção facial na imagem.

O desenvolvimento de técnicas para detecção facial começou por volta de 1990 (GRGIC MISLAV; DELAC, 2007).

Dentre as técnicas pesquisadas, pode-se dividi-las em três categorias diferentes:

- Detecção por Cor;
- Detecção por Movimento;
- Detecção em ambientes não controlados.

A detecção por cor leva em consideração os valores de RGB (ou valores em outro sistema de cor) da pele para encontrar áreas onde há uma maior probabilidade de se existir um rosto. Através do uso de filtros e *thresholds*, é possível criar uma imagem binária onde as regiões de pele recebem o valor 255 (branco, na escala de cinza) e o resto recebe 0 (preto, na escala de cinza). Desta forma, é possível analisar quais regiões possuem dois “buracos”, onde, em geral, indicam os olhos (CHANG; ROBLES, 2000). Este é um método simples, rápido, mas pouco robusto, uma vez que existe uma grande variedade de tons de pele e as diferentes condições de luz podem afetar a detecção das regiões de pele.

Já a detecção por movimento parte do pressuposto de que um rosto está quase sempre em movimento (raramente estará estático diante de uma câmera). Desta forma, é possível analisar quais partes de uma imagem mudam de um frame para o outro, sendo isto o indicativo do movimento do rosto (REIGNIER, 1995). Entretanto, esta técnica será falha caso haja mais de um objeto se movendo na tela. Para reduzir esse erro, é possível utilizar o fato de que os olhos humanos, em geral, piscam simultaneamente. Havendo duas áreas de movimento com uma distância aceitável entre si (i.e. próxima a distância média entre os olhos de uma rosto), é possível ter uma maior precisão da detecção das faces (PAN et al., 2007) (FRISCHHOLZ, 2014).

Por fim, a detecção em ambientes não controlados, um dos maiores e mais complexos objetivos a ser alcançado na área, já foi explorado através da utilização de técnicas

de inteligência artificial, como redes neurais e *Support Vector Machines* (SVM) (SHAN, 2010). Essas técnicas apresentam grande precisão nas detecções, mas podem ser lentas para aplicações em tempo real.

Dentro desta última categoria, há também os classificadores em cascata de Viola & Jones, que utilizam simples aspectos para caracterizar um rosto (VIOLA; JONES, 2004). Este método representou uma revolução na história da detecção de objetos (não somente rostos) por ter sido o primeiro rápido o suficiente para as aplicações em tempo real. Os classificadores são treinados através da técnica *AdaBoost*, empregando os *Haar-wavelets features*. O *AdaBoost* (SCHAPIRE; SINGER, 1999) é uma das mais bem sucedidas técnicas de aprendizado de máquinas aplicado na visão computacional. Posteriormente, eles estenderam a detecção com *Haar-like features* rotacionados com diferentes algoritmos de *boosting* (LIENHART; KURANOV; PISAREVSKY, 2003).

Necessário para a compreensão da metodologia posteriormente, segue-se um maior detalhamento sobre esse tipo de detector.

1.3.1 Classificadores Haar em Cascata

O método proposto por Viola & Jones consiste em utilizar um *framework* de detecção baseado em *features*, ou seja, características simples, reminiscentes das funções baseadas no Haar, utilizadas por (PAPAGEORGIOU; OREN; POGGIO, 1998). O principal motivo em se utilizar *textitfeatures* ao invés de simples valores de pixels é o de reduzir as variações dos dados dentro de uma classe e aumentar a variação entre os dados de duas classes diferentes, fazendo a classificação mais simples.

Viola & Jones utilizaram em seus trabalhos três tipos de *features*: Com dois, três e quatro retângulos. Posteriormente, (LIENHART; KURANOV; PISAREVSKY, 2003) estenderam esses *features* à rotação.

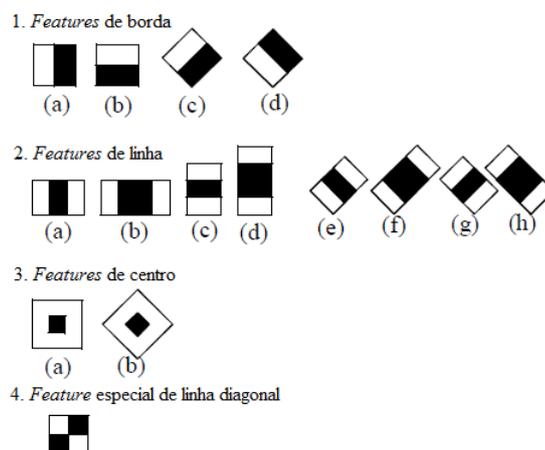


Figura 3 – Haar *features*

Fonte: (OPENCV, 2014a)

O valor de um *feature* é igual a diferença entre as somas dos pixels nas áreas pretas e a soma dos pixels nas áreas brancas do mesmo. Para os *features* originalmente propostos por Viola & Jones, a função *Summed Area Table* (Tabela de Área Somada) é utilizada para calcular o valor:

$$SAT(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (1.1)$$

Para os *features* rotacionados, a função *Rotated Summed Area Table* (Tabela da Soma da Área Rotacionada) é utilizada:

$$RSAT(x, y) = \sum_{y' \leq y, y \leq y' - |x - x'|} I(x', y') \quad (1.2)$$

Quanto maior o valor do *feature*, ou seja, maior a diferença entre a soma dos pixels nas áreas pretas e a soma dos pixels nas áreas brancas, tem-se um maior casamento daquela região da imagem com o *feature*.

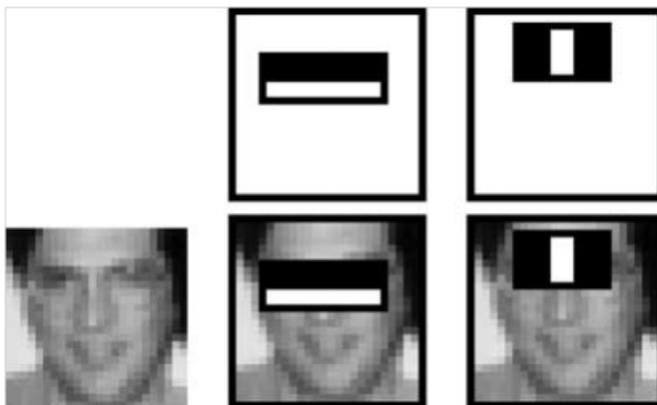


Figura 4 – *features* em um rosto

Fonte: (VIOLA; JONES, 2004)

Features retangulares são bem sensíveis a presença de bordas, barras e outras estruturas simples em uma imagem, mas são simples quando comparados a filtros direcionais (FREEMAN; ADELSON, 1991), pois as únicas direções que eles podem assumir são vertical, horizontal e diagonal. Para conseguir uma melhor generalização, esses *features* são modificados em tamanho e proporções de forma a conseguir uma representação mais completa de uma imagem. Empiricamente, nota-se que os *features* retangulares conseguem prover uma representação rica de uma imagem, o que suporta um aprendizado eficaz. Além disso, o gasto computacional para se calcular estes valores é extremamente pequeno, quando comparado aos filtros direcionais, e isso por si só compensa a simplicidade deles.

Tendo-se um conjunto de imagens positivas e negativas do objeto que se quer detectar, é possível treinar um classificador. Viola & Jones utilizaram o método de otimização

de *AdaBoost* (FREUND; SCHAPIRE, 1995) para a selecionar os *features* e para realizar o treinamento dos classificadores. Na sua forma original, o algoritmo de *AdaBoost* é usado para aumentar a performance da classificação de um simples algoritmo de aprendizado (por exemplo, um *perceptron*). Ele faz isso combinando uma coleção de funções de classificadores fracos para conseguir um classificador mais forte.

O termo cascata se refere a uma árvore de classificação degenerada, onde cada estágio é treinado para detectar quase todos os objetos de interesse (o rosto, por exemplo) enquanto rejeita uma determinada fração de padrões de não-objetos.

1.4 Reconhecimento Facial

Sabendo onde o rosto está, pode-se então fazer o reconhecimento do mesmo. O reconhecimento consiste em determinar uma identidade a uma amostra, dada uma base de treinamento. Apesar de parecer um processo fácil, este problema apresenta diversas dificuldades, a citar:

1. É possível que na hora da obtenção da amostra de teste, a pessoa esteja em uma posição diferente das amostras de treinamento.
2. Existe a dificuldade em se definir o quão similar uma amostra deve ser para ser considerada como parte da classe.

A forma mais simples de comparar duas imagens é fazer uma comparação direta pixel a pixel e aplicar um filtro. Os pixels iguais recebem 1 e os diferentes recebem zero. Tira-se o módulo de todos os pixel filtrados. Quanto mais próximo de 1, mais similares são as imagens.

Entretanto, devido ao problema (1), é bem provável que a identificação retorne falsa, mesmo que as duas amostras pertençam a mesma pessoa. Uma forma de tentar amenizar este problema é tirar uma média das imagens da database e fazer a comparação da imagem de prova com essa imagem média. Ainda assim, devido à grande complexidade do rosto humano, esse método pode falhar.

Sendo assim, é necessário encontrar os *features* mais importantes no rosto (como os olhos, o nariz, a boca, as distâncias entre cada um deles etc) e fazer a comparação destas características. Lembrar que enquanto na detecção, estes *features* representam olhos, narizes, bocas etc de uma forma geral (como o nariz de um ser humano deve ser), para o reconhecimento, esta caracterização deve ser mais específica de forma a representar apenas um indivíduo.

O modo mais simples de se caracterizar uma imagem é extrair seus valores de níveis de cinza. Dependendo do pré-processamento a ser realizado e o método para realizar o reconhecimento, eles podem retornar bons resultados, como pode ser visto em (HEISELE; HO; POGGIO, 2001).

Entretanto, dependendo das condições de iluminação do local, das diferentes poses que uma pessoa pode exibir em frente a câmera, a utilização de valores de cinza pode não caracterizar suficientemente um rosto, levando a muitos erros de identificação. Desta forma, vê-se que o primeiro passo no reconhecimento é conseguir um bom conjunto de *features*.

Em posse destes, é necessário então encontrar um bom método de reconhecimento. Em geral, os métodos de reconhecimento são, na realidade, métodos de classificação onde estes apenas verificam qual classe a amostra a ser verificada está mais próxima. Ou seja, se testarmos uma amostra de um rosto que não esteja presente no banco de dados utilizado para o treinamento, este “reconhecedor” irá retornar apenas a classe a qual ele mais se aproxima e não dizer que a pessoa é desconhecida.

Desta forma, nos deparamos com a definição de um problema de *open set*. Em geral, os problemas de classificação possuem amostras que identificam inteiramente as classes a serem diferenciadas. Entretanto, no caso do reconhecimento facial, só se tem amostras da classe positiva (a pessoa), enquanto que da classe negativa só temos um pequeno número do total que representa a classe (todas as outras pessoas existentes no mundo). Desta forma, é importantíssima a obtenção de características que representem bem as classes positivas, de forma a se poder distinguir pessoas que não pertencem a base de dados de treinamento.

1.4.1 Extratores de Características

Existem diversos métodos para extração e comparação de características de imagens. Em geral, estas características representam locais da imagem onde há um grande contraste. Desta forma, a boa detecção dos descritores depende diretamente na qualidade e iluminação da imagem.

Outra forma de se conseguir essas características é fazer uma decomposição das imagens para um subspaço com menos dimensões, o que facilita a posterior comparação. Se considerarmos cada pixel como uma característica válida para o reconhecimento, em uma imagem de 100x100 teremos 10000 características a comparar. Entretanto, nem todas essas características são úteis e desta forma é interessante encontrar quais as de maior peso na identificação.

Para isto, existe a decomposição e análise de componentes principais, em inglês, *Principal Components Analysis* (PCA). Para o propósito de identificação facial se dá o nome de *Eigenfaces*. Este método foi proposto para transformar um conjunto de variáveis correlacionadas em um conjunto menor de variáveis não correlacionadas. A ideia é que um conjunto de dados é normalmente descrito por variáveis correlacionadas e portanto poucas realmente retêm as informações da imagem. O método PCA encontra as direções com maior variância nos dados, chamados componentes principais. Inicialmente proposto por

(SIROVICH; KIRBY, 1987), desde então diversas variantes foram desenvolvidas como o *Adaptively Weighted Sub-Pattern PCA* (TAN; CHEN, 2005).

Entretanto, este método não considera nenhuma classe, verificando apenas quais características mais diferenciam as amostras entre si como um todo, o que pode acabar resultando em perda de informação.

Tentando resolver este problema, a Análise Linear Discriminativa, em inglês, *Linear Discriminative Analysis* (LDA), mais conhecida como *Fisherfaces*, inicialmente proposto por (FISHER, 1936) para o reconhecimento de flores, foi aplicada para o reconhecimento facial. Para encontrar uma combinação de características que melhor separam um conjunto em classes, esta análise maximiza o raio entre as classes, enquanto minimiza os raios das mesmas. Isso vem do raciocínio de que imagens de mesma classe devem se agrupar, enquanto classes diferentes devem estar o mais distantes possível das outras (BELHUMEUR; HESPANHA; KRIEGMAN, 1997).

Em especial, vamos falar dos Padrões Binários Locais, em inglês, *Local Binary Patterns* (LBP). Eles possuem uma vantagem sobre os métodos apresentados até então devido ao fato de serem invariantes às mudanças monotônicas dos níveis de cinza em uma imagem, o que pode ser visualizado na seguinte imagem:

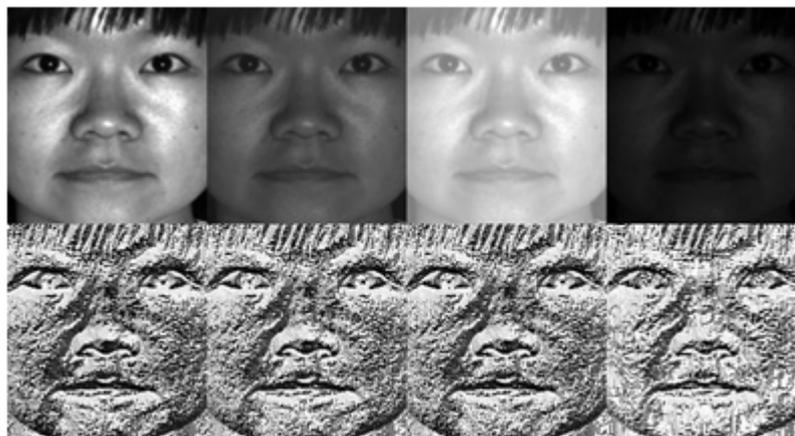


Figura 5 – LBP para imagens com mudanças monotônicas de cinza.

Fonte: (OPENCV, 2014b)

1.4.1.1 Padrões Binários Locais

Os Padrões Binários Locais são encontrados através de um operador que transforma a imagem em um vetor ou matriz com valores que descrevem a aparência, em pequena escala, da imagem original. Esses valores, ou a estatística desses valores, normalmente o histograma, são então utilizados como características para o reconhecimento (PIETIKAINEN et al., 2011).

O método foi originalmente proposto por (OJALA; PIETIKÄINEN; MÄENPÄÄ, 2001) e se baseia na análise da textura 2D de uma imagem. A principal ideia do LBP é

resumir a estrutura local de uma imagem ao comparar o valor da intensidade de cinza de um pixel com os valores de seus vizinhos. Se a intensidade do pixel central for maior ou igual ao valor do pixel vizinho, o padrão recebe 1, caso contrário, recebe 0. Ao fim desse processo, para cada pixel, terá um número binário. Se considerarmos uma vizinhança de 8 pixels, o número binário terá 8 dígitos e existirá 2^8 possíveis combinações, estas chamadas de Padrões Binários Locais.

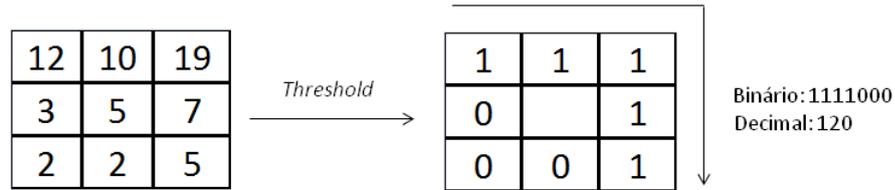


Figura 6 – Padrão Binário Local.

Fonte: Autoria própria.

Uma forma mais formal de se descrever os LBPs é através do seguinte operador:

$$LBP(x_c, x_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c) \quad (1.3)$$

Onde:

- (x_c, y_c) representa a posição do pixel central;
- i_c é o valor da intensidade de cinza do pixel central;
- i_p é o valor da intensidade de cinza do pixel vizinho p ;
- P é o número de pixels vizinhos.

E s é a função de sinal representada a seguir:

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (1.4)$$

Esse tipo de descrição consegue capturar de forma detalhada importantes padrões para o reconhecimento de uma imagem como:

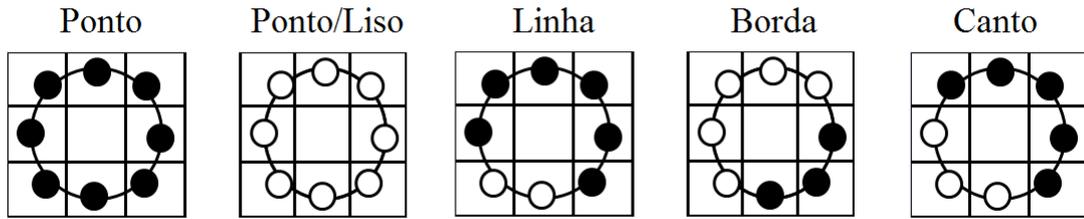


Figura 7 – Padrões encontrados pelo LBP.

Fonte: Autoria própria.

Entretanto, o operador utilizando uma vizinhança fixa de 8 pixels acaba falhando na captura de detalhes com escalas maiores. Desta forma, posteriormente, (OJALA; PIETIKÄINEN; MÄENPÄÄ, 2002) propôs o cálculo dos LBPs com um raio variável por meio da seguinte equação:

$$y_p = x_c + R \cos\left(\frac{2\pi p}{P}\right) \tag{1.5}$$

$$x_p = y_c + R \sin\left(\frac{2\pi p}{P}\right) \tag{1.6}$$

Onde R é o raio que se deseja abranger.

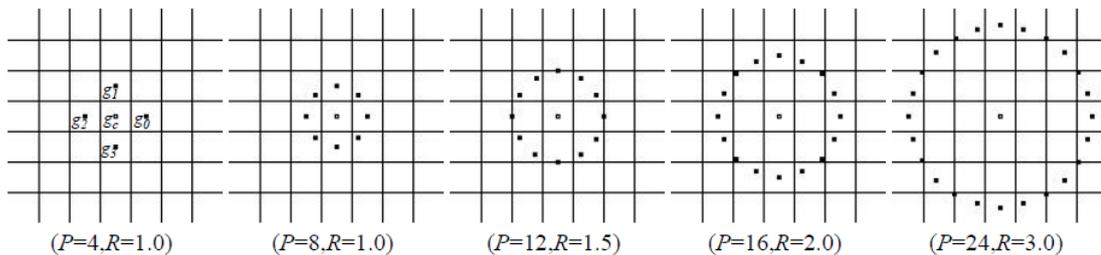


Figura 8 – Vizinhança com diferentes tamanhos de R.

Fonte: (OJALA; PIETIKÄINEN; MÄENPÄÄ, 2002)

Pontos cujas coordenadas se encontrem fora dos limites da imagem são interpolados.

1.4.2 Máquinas de Vetores de Suporte

Máquina de Vetores de Suporte, também conhecida como SVM (*Support Vector Machine*), é uma técnica originalmente apresentada por (VAPNIK, 2000) que incorpora a Minimização de Risco Estrutural (*Structural Risk Minimisation - SRM*). O SRM se mostrou superior à tradicional Minimização de Risco Empírico (*Empirical Risk Minimisation - ERM*), pois reduz a barreira superior no erro esperado, em oposição ao EMR que reduz o erro nos dados de treinamento. Tentar reduzir o erro em cima das amostras de treinamento pode levar ao chamado *over fitting*, ou seja, especialização demasiada em cima das

amostras de treinamento, tornando um modelo pouco genérico. O SVM realiza a separação entre amostras de duas classes diferentes por meio da projeção dessas amostras em um espaço dimensionalmente maior, traçando um hiperplano que consiga separar, se não todas, a maior parte dessas amostras, maximizando a distância entre as amostras de cada classe e o hiperplano.

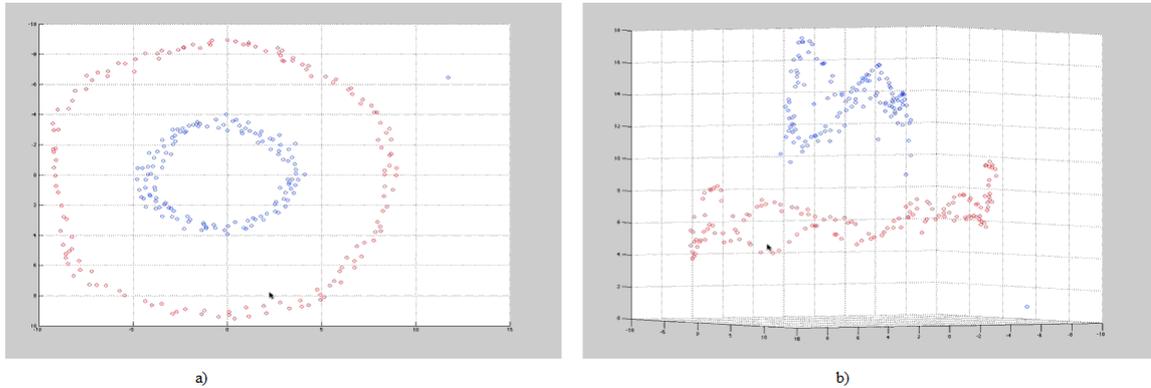


Figure 9 – Separação dos dados pelo SVM. a) Dados não linearmente separáveis. b) Ao jogar esses dados para um espaço de maior dimensionalidade, é possível separar estes dados por meio de um hiperplano.

Fonte: Autoria própria.

1.4.2.1 Classificação por Vetores de Suporte

Considerando um problema de classificação com duas classes, com um conjunto de amostras de treinamento $(x_1, y_1), \dots, (x_m, y_m)$, onde $x_i \in R^n$ é um vetor de *features*, dá-se uma *label*, ou seja, uma etiqueta para cada amostra $y_i \in [+1, -1]$. O SVC separa essas duas classes através de um hiperplano $W \bullet X + b = 0$. Para uma das modalidades de SVC, o C-SVC, resolve-se o seguinte problema de otimização primal:

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \omega^T \omega + C \sum_{i=1}^l \xi_i \\ \text{sujeito a} \quad & y_i (\omega^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \dots, l \end{aligned} \quad (1.7)$$

Onde $\phi(x_i)$ mapeia x_i para um espaço de dimensão maior e o $C > 0$ é um parâmetro de regularização. Existe a possibilidade de existir uma grande dimensionalidade no vetor ω e, desta forma, em geral se resolve este problema dual da seguinte forma:

$$\min_{\alpha} \quad \frac{1}{2} \alpha^T Q \alpha - r^T \alpha$$

$$\begin{aligned} \text{sujeito a } \quad & y^T \alpha = 0 \\ & 0 \leq \alpha_i \leq C, i = 1, \dots, l \end{aligned} \quad (1.8)$$

Onde $r = [1, \dots, 1]^T$ é um vetor composto somente de 1, Q é uma matriz positiva de dimensão $l \times l$, $Q_{ij} \equiv y_i y_j K(x_i, y_i)$, e $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ é a função Kernel. Ele é o responsável por traçar o hiperplano no espaço dimensionalmente maior, para a divisão das amostras entre as classes e, dependendo de sua forma, o SVC pode separar tanto amostras linearmente separáveis quanto as não linearmente separáveis. Qualquer função que designe um plano pode ser utilizada como um Kernel, mas dentre as mais utilizadas, pode-se citar o Kernel linear, polinomial, função de base radial (RBF), sigmoide, como pode ser vistos nos trabalhos de (CHANG; LIN, 2011), (OSUNA; FREUND; GIROSI, 1997).

Uma vez que o problema dual é resolvido, o hiperplano ω ótimo satisfaz:

$$\omega = \sum_{i=1}^l y_i \alpha_i \phi(x_i) \quad (1.9)$$

E a função de decisão é:

$$\text{sgn}(\omega^T \phi(x) + b) = \text{sgn}\left(\sum_{i=1}^l y_i \alpha_i K(x_i, x) + b\right) \quad (1.10)$$

1.4.2.2 Regressão por Vetores de Suporte

Os SVMs além de serem utilizados para classificação também são utilizados para regressão. A Regressão por Vetores de Suporte, também conhecida como SVR (*Support Vector Regression*) tem como objetivo encontrar um vetor que consiga interpolar um conjunto de dados, em um espaço de maior dimensionalidade.

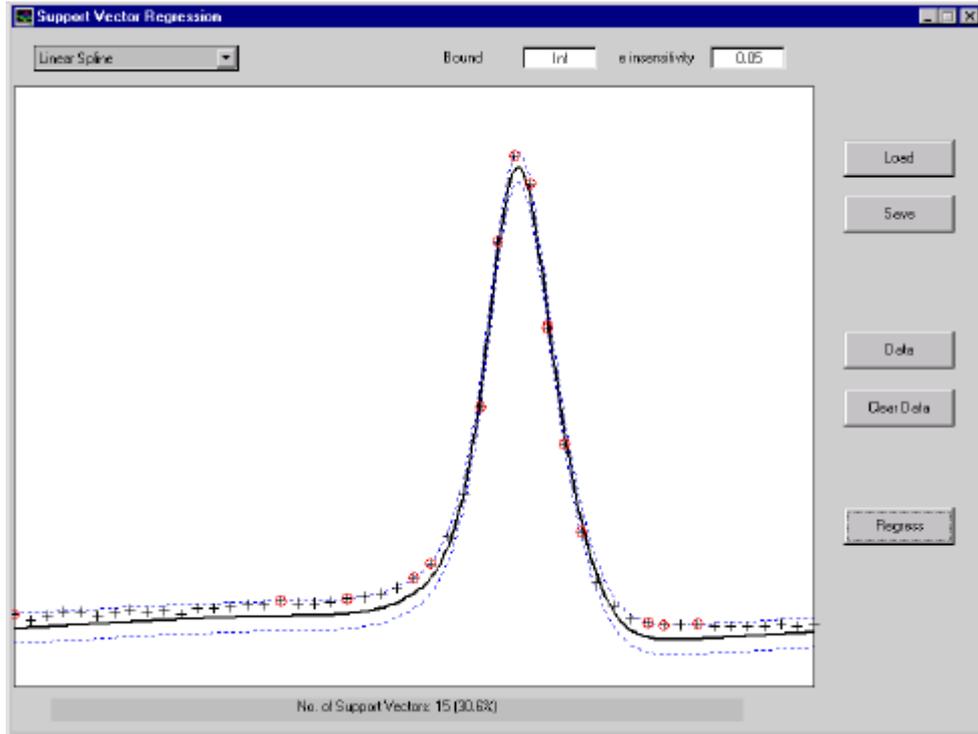


Figura 10 – Exemplo de Regressão por Vetores de Suporte.

Fonte:(GUNN et al., 1998)

Considerando um conjunto de pontos de treinamento $\{(x_1, z_1), \dots, (x_l, z_l)\}$, onde $x_i \in R^n$ é um vetor de *feature* e $z_i \in R^l$ é a saída desejada. Sob os parâmetros dados $C > 0$ e $\epsilon > 0$, a forma padrão da regressão por vetores de suporte, de acordo com (VAPNIK; VAPNIK, 1998):

$$\begin{aligned}
 \min_{\omega, b, \xi, \xi^*} \quad & \frac{1}{2} \omega^T \omega + C \sum_{i=1}^l \xi_i + C \sum_{i=1}^l \xi_i^* \\
 \text{sujeito a} \quad & \omega^T \phi(x_i) + b - z_i \leq \epsilon + \xi_i, \\
 & z_i - \omega^T \phi(x_i) - b \leq \epsilon + \xi_i^*, \\
 & \xi_i, \xi_i^* \geq 0, i = 1, \dots, l.
 \end{aligned} \tag{1.11}$$

O problema dual é:

$$\begin{aligned}
 \min_{\alpha, \alpha^*} \quad & \frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \epsilon \sum_{i=1}^l (\alpha + \alpha^*) + \sum_{i=1}^l z_i (\alpha - \alpha^*) \\
 \text{sujeito a} \quad & e^T (\alpha - \alpha^*) = 0 \\
 & 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, \dots, l.
 \end{aligned} \tag{1.12}$$

Onde $Q_{ij} = K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$.

E depois de resolver o problema dual, a função de aproximação é:

$$\sum_{i=1} l(-\alpha_i + \alpha_i^*) K(x_i, x) + b. \quad (1.13)$$

1.5 Medidas Básicas de Avaliação de Performance de Classificadores

No âmbito da inteligência artificial e do aprendizado de máquinas, existe uma grande variedade de medidas utilizadas para avaliar o poder de predição dos sistemas. Dependendo dos dados, algumas medidas são mais apropriadas que outras e é importante entender o que cada uma significa para uma correta avaliação.

É necessário, portanto, conhecer algumas terminologias. A seguir, uma matriz confusão que mostra como as predições de instâncias são tabuladas:

		<i>Label</i> prevista	
		Positivo	Negativo
<i>Label</i> conhecida	Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Figura 11 – Matriz Confusão.

A partir desses valores na matriz confusão, as seguintes medidas podem ser calculadas:

- **Precisão**, que indica a porcentagem de amostras positivas classificadas corretamente sobre o total de amostras classificadas como positivas;

$$Precisão = \frac{VP}{VP + TN} \quad (1.14)$$

- **Sensibilidade**, que indica a porcentagem de amostras positivas classificadas corretamente sobre o total de amostras positivas;

$$Sensibilidade = \frac{VP}{VP + FN} \quad (1.15)$$

- **Especificidade**, que indica a porcentagem de amostras negativas identificadas corretamente sobre o total de amostras negativas;

$$Especificidade = \frac{VN}{VN + FP} \quad (1.16)$$

- **Acurácia**, que indica a porcentagem de amostras positivas e negativas classificadas corretamente sobre a soma de amostras positivas e negativas;

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (1.17)$$

É importante levar em consideração todos esses quatro fatores na hora de avaliar um classificador devido ao fato da possibilidade de desbalanceamento da base de dados de teste (muitas amostras positivas, poucas amostras negativas, por exemplo). Apesar de ser uma informação relevante, a precisão nem sempre pode ser utilizada como parâmetro de avaliação do classificador, pois em base dados desbalanceadas com um maior número de amostras negativas, a precisão pode retornar um número baixo, mesmo que a classificação esteja correta. Os parâmetros que mais podem informar sobre o desempenho do classificador é a sensibilidade e a especificidade.

Desta forma, um gráfico muito utilizado na avaliação de desempenho de classificadores é o chamado curva ROC (*Receiver Operating Characteristic*) e este mostra um enfoque gráfico da compensação entre as taxas de verdadeiros positivos e falsos positivos. Cada ponto na curva corresponde a um dos modelos induzidos pelo classificador e um bom modelo deve estar próximo do ponto (0,1).

Modelos que estão localizados em cima linha diagonal do gráfico são modelos aleatórios. Bons modelos possuem sua curva acima desta linha.

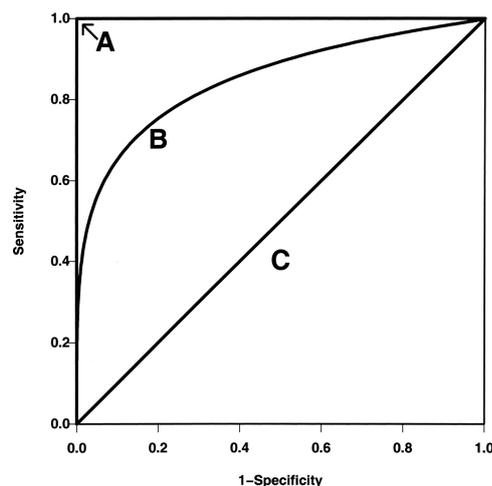


Figura 12 – Exemplo de curva ROC. A) Curva Ideal. B) Bom Teste. C) Curva randômica.

Fonte: (ZOU; OMALLEY; MAURI, 2007)

1.6 Hardware

1.6.1 Microcontroladores

Um microcontrolador é um dispositivo composto de um processador, memórias e periféricos de entrada e saída. É importante salientar a diferença entre um microcontrolador e um microprocessador. Enquanto o microcontrolador tem autonomia para funcionar, precisando apenas de um programa e dispositivos de entrada e saída, o microprocessador é apenas a unidade central de processamento, sendo necessário a conexão com memórias, barramentos de entrada e saída etc (ZELENOVSKY; MENDONÇA, 2005)

1.6.1.1 Processador

O processador é o responsável pela inteligência do microcontrolador, sendo encarregado de tomar as decisões através da execução de um programa. É constituído de três unidades diferentes: Unidade lógica e aritmética, registradores internos e um bloco de temporização e controle.

A primeira é responsável pela execução das instruções correspondentes às operações aritméticas e lógicas. Ao executar uma instrução de uma dessas operações, ela também define a condição numérica do resultado através de alguns bits sinalizadores.

Os registradores internos são interligados entre si em paralelo, sendo o número de registradores e o número de bits que eles suportam dependentes das características do microcontrolador/microprocessador. Eles são formados por flip-flops e, desta forma, eles podem realizar operações de leitura e escrita.

Como última unidade do processador, o bloco de temporização e controle é responsável pelo controle de fluxo de informações do microprocessador para as unidades de entrada e saída e para as unidades de memória. É ele quem define os sinais de temporização para a sincronia do fluxo de informações dentro do sistema, por meio de definição de sinais de leitura (*read*), escrita (*write*), inicialização do sistema (*reset*), entre outros

1.6.1.2 Memórias

As unidades de memórias são divididas em duas: Memória de armazenamento do programa e memória de armazenamento de informações.

A memória de armazenamento de programa é responsável por guardar o programa que definirá as tarefas que a CPU deve executar. É uma memória do tipo não-volátil (a informação não se perde quando o dispositivo é desenergizado) devido ao fato de que toda vez que o microcontrolador for energizado, ele irá procurar pelo programa. Se não existir programa, o microcontrolador não funcionará.

A memória de armazenamento de informações são do tipo volátil, conhecida como memória de acesso randômico, em inglês, *Random Access Memory* (RAM). Ela permite a leitura e escrita de informações que são geradas e necessárias durante a execução das tarefas definidas pelo programa do microcontrolador.

1.6.1.3 Unidade de entrada e saída

É conhecida como unidade de I/O e é responsável pela ligação entre o microcontrolador e o mundo externo. Em geral, são circuitos integrados que conseguem ler e armazenar informações vindas de dispositivos externos, tais como teclados, sensores, canais de comunicação serial, além de conseguirem também enviar informações do microcontrolador para o mundo externo, através de diodos emissores de luz, em inglês, *Light Emitting Diode* (LED), *displays*, autofalantes etc.

1.6.2 Tecnologia ZigBee

ZigBee é uma tecnologia desenvolvida como um padrão global aberto para atender as necessidades de baixo custo e baixo consumo que redes M2M (*Machine to machine*, redes que permitem a comunicação entre dispositivos do mesmo tipo) exigem. O padrão *ZigBee* opera nas especificações físicas de rádio IEEE 802.15.4 e opera em bandas não licenciadas como 2,4 GHz, 900 MHz and 868 MHz (BARONTI et al., 2007), com alcance entre 10 e 100 metros. A especificação 802.15.4 foi ratificada em 2003 pelo Instituto de Engenheiros Elétricos e Eletrônicos (IEEE). Esta especificação é um protocolo de rádio baseado em pacotes, direcionado para dispositivos de baixo custo e que são alimentados por baterias.

Dentre as muitas características desta tecnologia, pode-se citar:

- O suporte para diversos tipos de topologia de rede, como ponto-a-ponto, ponto para vários pontos, redes em malha;
- Um baixo ciclo de trabalho, que provê o aumento da vida da bateria;
- Baixa latência;
- Suporte a até 65000 nós por rede;
- Criptografia de 128-bit AES para conexões de dados seguras;
- Prevenção de colisão;

Desta forma, esta tecnologia é amplamente utilizada em aplicações embarcadas que requerem baixas taxas de transmissão de dados e baixo consumo de energia. Exemplos de caso de uso, podem-se citar a domótica, automação empresarial, coleta de dados médicos, redes sem fio de sensores, entre outros.

1.6.2.1 Tipos de dispositivos ZigBee

Eles podem ser de três tipos (LIU; ZHANG; DING, 2014):

- *ZigBee* coordenador (ZC): O dispositivo com maior capacidade, o coordenador forma a raiz da árvore da rede e pode servir como ponte para outras redes. Existe apenas um *ZigBee* coordenador dentro de uma rede, uma vez que ele é o dispositivo que originalmente começou a rede. Ele guarda as informações sobre a rede, agindo como central de confiabilidade e como repositório para as chaves de segurança.
- *ZigBee* roteador (ZR): Além de executar uma função de aplicação, o ZR pode agir como um roteador intermediário, passando dados para outros dispositivos.
- *ZigBee* dispositivo final (ZED): Contém apenas a funcionalidade necessária para conversar com o nó paterno (que pode ser um ZC ou um ZR). Ele não pode chavear

dados para outros dispositivos. Este tipo de relação permite que o nó possa dormir durante a maior parte do tempo, reduzindo o consumo de energia. Um ZED requer a menor quantidade de memória e desta forma, pode ser mais barato que um ZC ou um ZR.

2 Metodologia

2.1 Sistema de Reconhecimento Facial - Funcionamento Geral

O Sistema de Reconhecimento Facial desenvolvido para este projeto tem como objetivo detectar um rosto em imagens capturadas por uma câmera e identificá-lo de acordo com rostos já registrados em uma base de dados.

Para o desenvolvimento deste sistema, consideraram-se as seguintes condições de contorno:

- a) A iluminação do local onde o sistema será implementado é controlada e constante;
- b) O local onde o sistema será implementado é estático;
- c) Detectar-se-á apenas rostos frontais;
- d) A câmera a ser utilizada para captura das imagens deve ter, no mínimo, 1 megapixel de resolução;
- e) O computador a ser utilizado para executar o programa tem como processador um intel®core i5, 4GB de memória RAM.

Além do programa, o sistema também engloba a comunicação via *ZigBee* com um sistema de acionamento de uma fechadura elétrica.

2.1.1 *Software*

Esse sistema pode operar em dois modos diferentes:

1. Modo de Aprendizado: Para registrar um rosto na base de dados;
2. Modo de Reconhecimento: Para reconhecer os rostos de acordo com a base de dados.

A seguir, serão apresentados o funcionamento de cada um desses modos.

2.1.1.1 Aprendizado

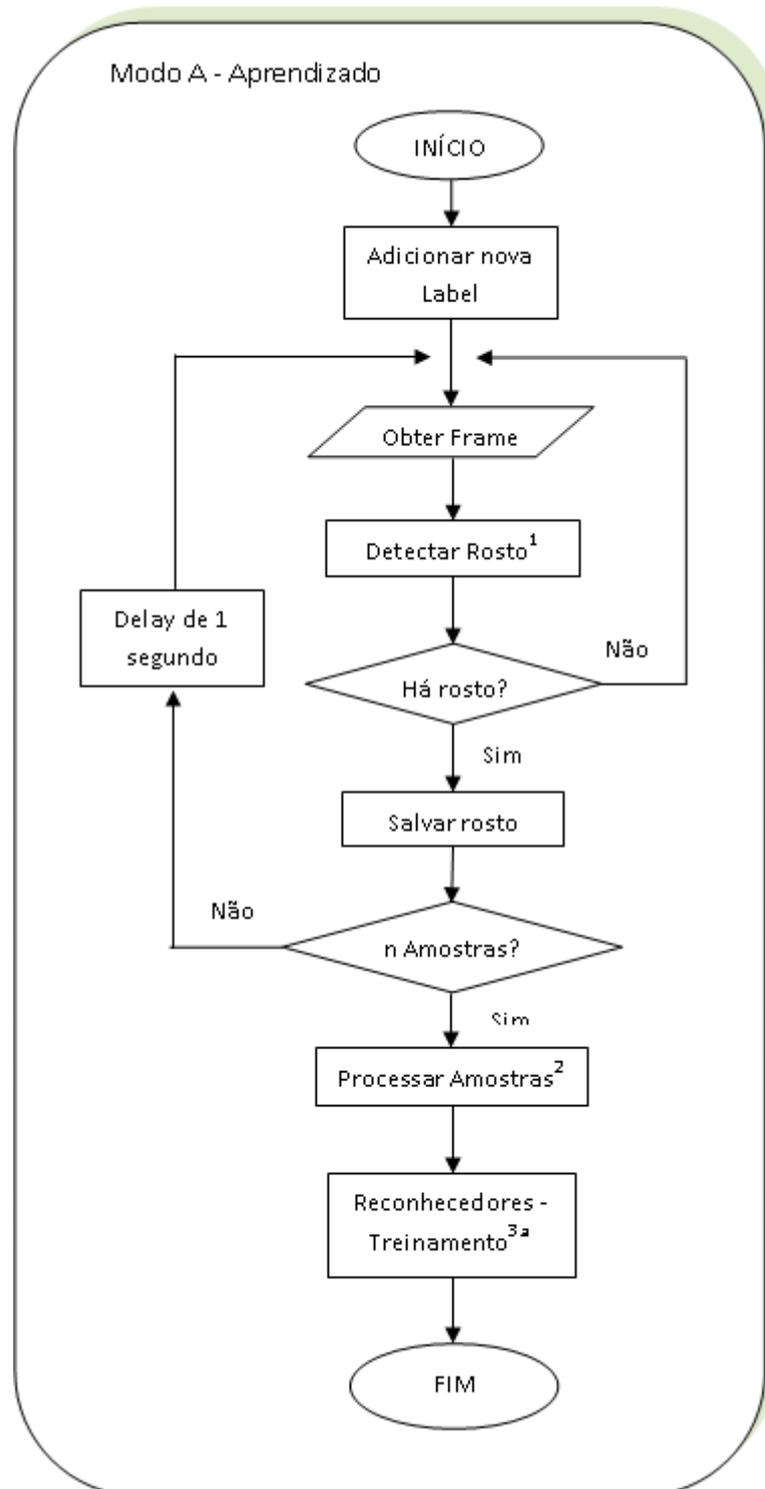


Figura 13 – Fluxograma A - Aprendizado.

Fonte: Autoria Própria.

Ao selecionar este modo, o programa criará uma nova *label* para o novo indivíduo a ser adicionado na base de dados. *Label*, neste contexto, é um número inteiro que irá identificar

cada um dos indivíduos dentro da base de dados. Captura-se um *frame*, ou quadro, da câmera, passando o mesmo pelo Detector de Rostos. É ideal que apenas a pessoa a ser registrada esteja na frente da câmera, mas caso mais de um rosto esteja dentro do alcance da mesma, apenas o maior rosto será processado. Se houver rosto no frame, o programa salva este rosto como uma amostra e repete o processo por *n* vezes. Entre cada amostra, espera-se um segundo para evitar o vício da base de dados com imagens muito similares.

Depois que as *n* amostras do indivíduo são adquiridas, as amostras são processadas, de forma a assumirem o formato necessário para o treinamento dos reconhecedores.

2.1.1.2 Reconhecimento

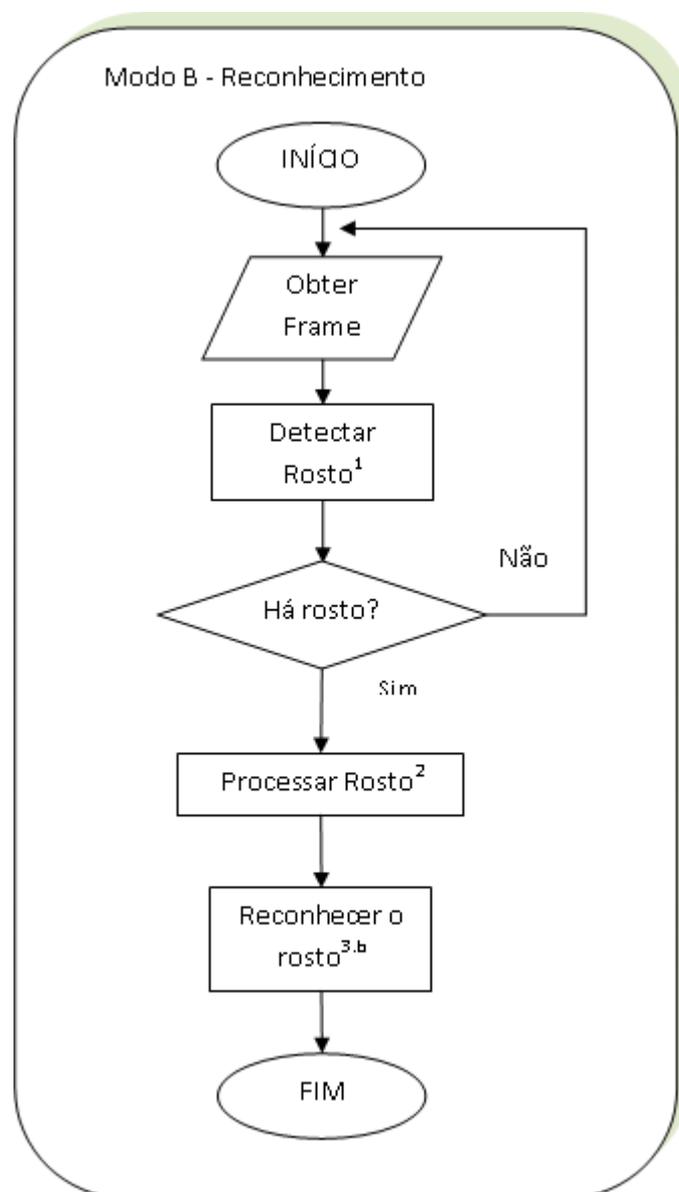


Figura 14 – Fluxograma B - Reconhecimento.

Fonte: Autoria própria.

Ao selecionar o modo de reconhecimento, o programa captura um quadro da câmera e o fornece como entrada para o Detector de Rostos. O Detector de Rostos irá verificar se há rosto ou não no frame. Havendo mais de um rosto, o detector irá considerar apenas o de maior área, retornando um **Objeto Face**. Ele é constituído de um retângulo, que indica a posição e área ocupada pelo rosto, e os pontos centrais dos elementos do rosto (olhos, nariz e boca). Caso não haja nenhum rosto, o programa pegará o próximo frame e repetirá a detecção de rosto.

Cada rosto retornado pelo Detector será processado de modo a se adequar ao formato necessário para que o Reconhecedor possa fazer a predição. Este processamento é o mesmo realizado no Modo Aprendizado. Em seguida, eles serão processados pelo Reconhecedor de Rostos que irá retornar o número da *label* do indivíduo, caso ele esteja cadastrado na base de dados, ou o valor -1 caso o indivíduo seja desconhecido.

2.1.2 Hardware

A parte física do sistema consiste basicamente em dois dispositivos *ZigBee*, um ligado ao computador, atuando como *ZigBee* coordenador; e um ligado a um microcontrolador, atuando como um *ZigBee* dispositivo final. Assim que o programa define a identidade da pessoa em frente a câmera, ele envia um sinal através do ZC para liberar a fechadura elétrica, caso a pessoa esteja registrada na base de dados.

Devido ao simples processamento de dados necessário para a aplicação, utilizou-se como microcontrolador da família *Arduino*. Os dispositivos *ZigBee* escolhidos foram as placas *XBee*© da *Digi International*. Para fazer a ligação do ZC com o computador, utilizou-se o *XBee Explorer*© da *SparkFun*, modelo S2, e para ligar o ZED com o microcontrolador, utilizou-se um *shield* do próprio fabricante do *Arduino*.

O programa estabelece uma comunicação serial com o ZC que por sua vez, se comunica de forma sem fio com o ZED. O ZED está ligado aos pinos de TX0 e RX0 do microcontrolador, e este possui um pequeno código em C para receber e enviar informações por estes pinos.

Para o acionamento da fechadura elétrica, o microcontrolador está ligado a um relé. Ele possui como entrada uma tensão de 5VDC/ 71mA e carga nominal de 10A 125VAC , 10A 24VDC. Uma vez que o Arduino só consegue suprir uma corrente de 40mA, é preciso utilizar um transistor bipolar, tendo-se o seguinte circuito:

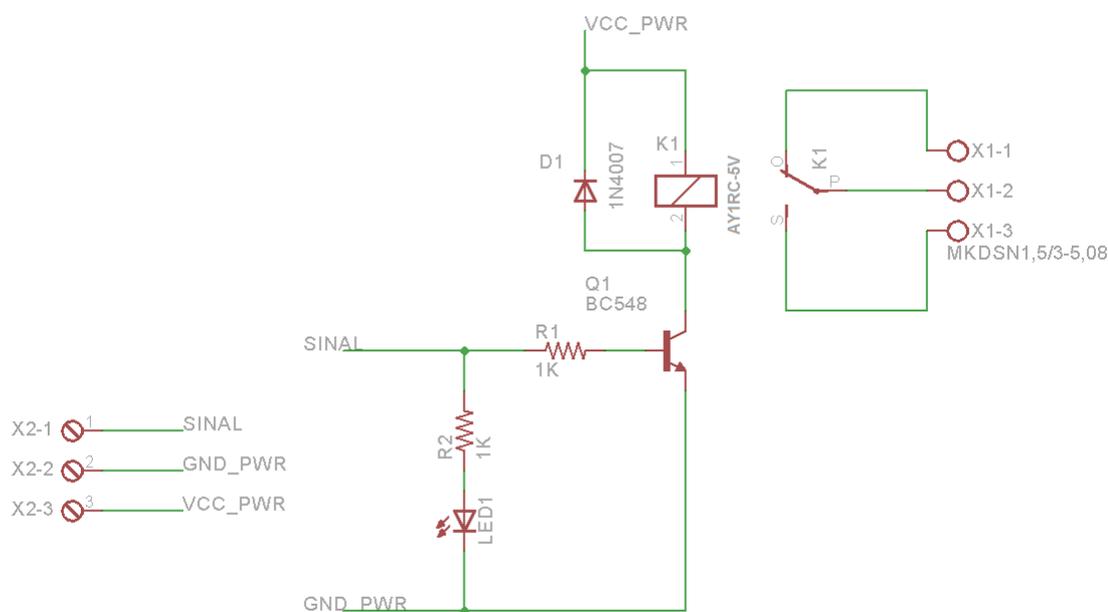


Figura 15 – Circuito de Acionamento da fechadura.

Fonte: Autoria Própria.

Os terminais X2-1, X2-2, X2-3 são ligados ao microcontrolador, onde o X2-1 é ligado ao pino que emite o sinal de liberação. Os terminais X1-2 é ligado a fechadura elétrica e o X1-3, ligado a fase da alimentação 12VAC vinda de um pequeno transformador AC/AC que converte 127/220V em 12V. Este circuito pode ser adquirido no formato de uma pequena placa. A fechadura elétrica fecha o circuito ligando seu outro terminal no neutro da fonte de alimentação:

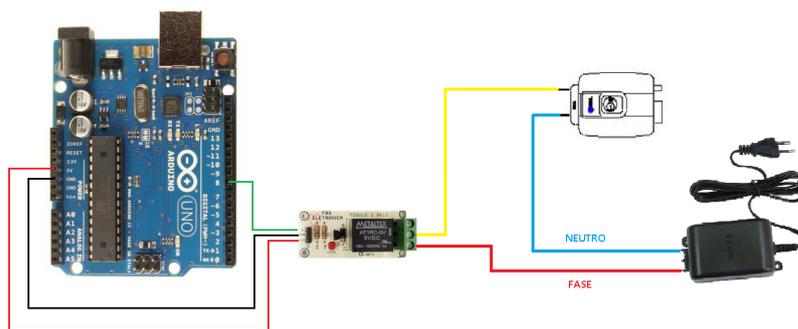


Figura 16 – Ligação da fechadura elétrica com o circuito de acionamento e o microcontrolador.

Fonte: (ARDUINO..., 2014; FBS..., 2014; HDL..., 2014)

As imagens são meramente ilustrativas e não representam fielmente as peças utilizadas no projeto (ver o Capítulo III).

2.2 Detector de Rostos

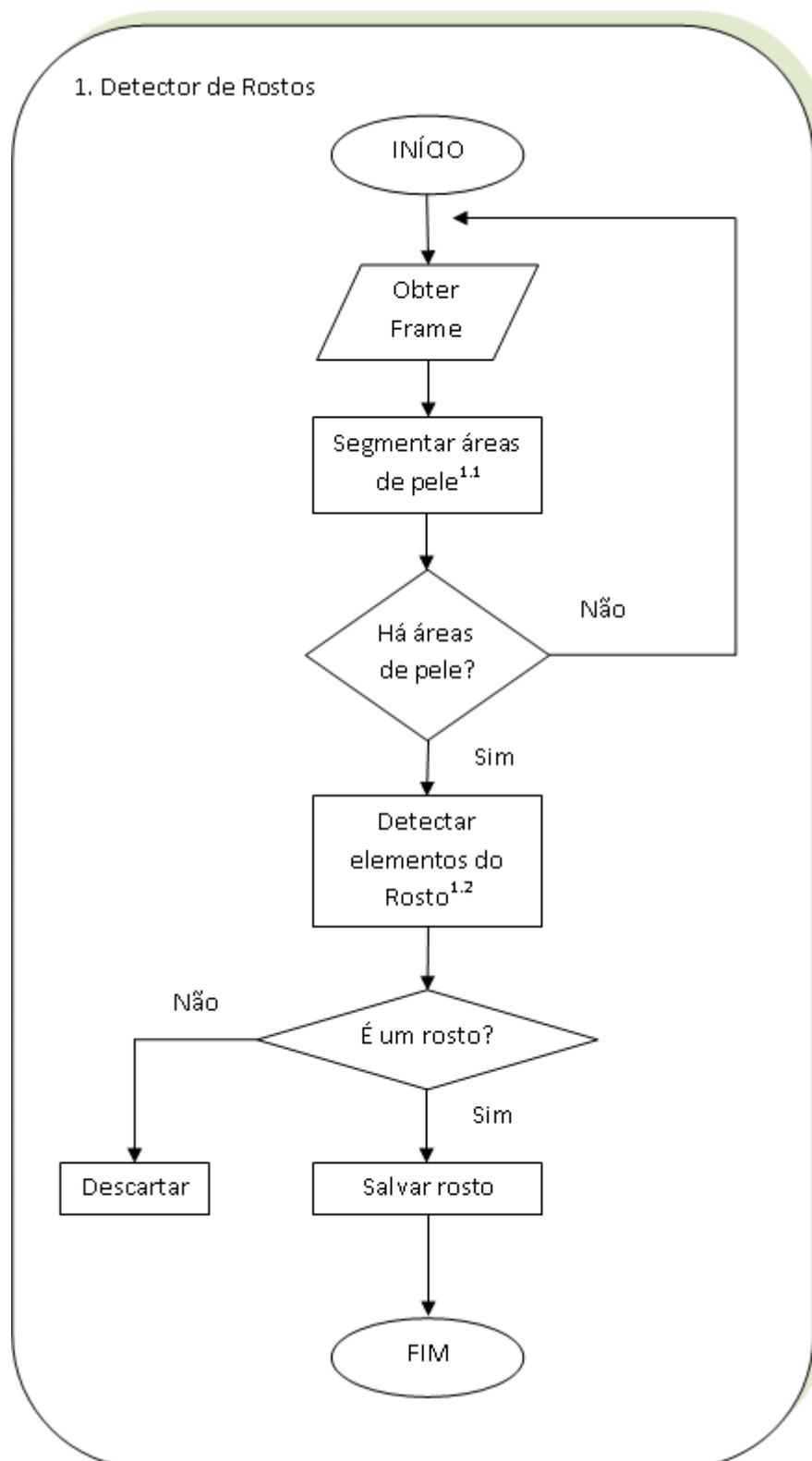


Figura 17 – Fluxograma 1 - Detector de Rostos.

Fonte: Autoria própria.

Em vista da aplicação deste sistema, é necessário colocar como prioridade o tempo de processamento das imagens. Um dos métodos mais utilizados para se detectar rostos é o chamado *Haar Cascade Classifier* ou Classificador Haar em Cascata, como já visto no referencial teórico.

No âmbito deste projeto, o Detector necessita retornar um objeto face, ou seja, o retângulo do rosto e os pontos centrais dos elementos do rosto (olhos, nariz e boca). Os Classificadores Haar em Cascata podem ser utilizados também para detectar esses elementos do rosto. Neste caso, deve-se, primeiramente, encontrar o retângulo do rosto e em seguida procurar pelos elementos dentro da área representada pelo retângulo.

Este processo, no entanto, se torna custoso. Para detectar os rostos, o Classificador Haar em Cascata procura por um conjunto de padrões que caracterizem um rosto. Esses padrões são os que representam os olhos, nariz e boca. Como o Classificador Haar em Cascata para o rosto só retorna o rosto, é preciso procurar por estes elementos novamente, de forma independente. Desta forma acaba-se procurando pelos elementos do rosto duas vezes.

De forma a verificar quanto tempo os classificadores Haar em Cascata leval para identificar o rosto e os elementos, realizou-se alguns testes em um computador com as mesmas características especificadas nas condições de contorno deste sistema. Ele apresentou, em média, um tempo de detecção de 8 a 10 ms para os elementos e para apenas o rosto, 313ms. No total, ele levou cerca de 360 ms para detectar tanto o rosto quanto os elementos.

Como se pode ver, a detecção dos elementos é relativamente mais rápida se comparada a detecção do rosto por inteiro. Além das razões já citadas, isso também se deve ao fato de que tentar encontrar uma correspondência de um rosto inteiro é algo muito mais complexo do que tentar encontrar uma correspondência de um único olho ou nariz.

Desta forma, pensou-se numa maneira de encontrar tanto o rosto quanto estes elementos com apenas uma busca para cada. A partir das várias técnicas de detecção de rosto pesquisadas, resolveu-se uní-las de modo a conseguir um método capaz de combinar os vários aspectos positivos de cada uma delas, sem que os pontos negativos afetassem demasiadamente os resultados.

É preciso, então, encontrar áreas onde a probabilidade de se ter um rosto seja grande. Tendo essas áreas, validá-las como rosto através da detecção dos elementos de rosto. Considerando que a maior parte de um rosto é composta por pele, a técnica de segmentação de pixels de pele se mostrou eficaz para esta tarefa.

Esta técnica retorna uma imagem filtrada onde os pixels podem assumir dois estados: Pele ou não pele. As áreas de pixels de pele são chamadas de áreas de pele e estas são consideradas como áreas de grande probabilidade de ser um rosto.

Analisando o problema empiricamente, vê-se que para validar essas áreas como um rosto, basta considerar que se houver dois buracos, ou seja, duas áreas de não pele, alinhados com o eixo menor da elipse aproximada da área de pele, tem-se um rosto. Esses

buracos correspondem aos olhos. Entretanto, caso a pessoa esteja de olhos fechados, esses buracos não existirão.

Sendo assim, para a verificação das áreas de pele, os classificadores Haar em cascata são mais apropriados, uma vez que já existem classificadores treinados para encontrar tanto olhos abertos como olhos fechados. Neste caso, extrai-se apenas as áreas de interesse na imagem original, correspondentes a essas áreas de pele, efetuando-se a detecção dos elementos nelas.

O método, então, ficou constituído dos seguintes passos:

1. Segmentar os pixels correspondentes à pele;
2. Em cada segmento, verificar a presença de olhos, nariz e boca, utilizando os detectores Haar em Cascata;
3. Caso estejam presentes e obedeçam às restrições geométricas, é um rosto;
4. Caso contrário, descartar o segmento.

2.2.1 Segmentação da Pele

Uma das principais tarefas no aprendizado visual é a construção de modelos estatísticos da aparência das imagens através de dados de pixel. Quando se têm poucas amostras para o treinamento, pode ser necessária a utilização de um algoritmo sofisticado de aprendizado para conseguir interoperar entre as amostras. Entretanto, se o número de amostras é grande, é possível criar algoritmos simples, computacionalmente eficiente para o aprendizado.

Considerando um grande número de amostras, modelos de histograma de cores se mostrou uma técnica eficiente para o aprendizado e classificação de pixels com cor de pele (KHAN et al., 2012).

Para conseguir segmentar os pixels correspondentes à pele, adotou-se a utilização de um Classificador Bayesiano com a técnica do histograma. Criou-se uma database com três classes de imagem:

1. Imagens apenas com pixels correspondentes à pele (12329 imagens de 20x20);
2. Imagens com pixels não correspondentes à pele (115 imagens de 1080x1920);
3. Imagens com pixels correspondentes e não correspondentes à pele (153 imagens de 1080x1920).

O motivo pelo qual as imagens com pixels correspondentes a pele terem resolução tão pequena é porque elas foram cortadas uma a uma manualmente de forma a conterem apenas pixels correspondentes a pele. Já as imagens para pixels não correspondentes à

pele são imagens de paisagem, sem a presença de qualquer pessoa. Por fim, as imagens correspondentes e não correspondentes à pele são fotos de pessoas em diversos lugares.

O espaço de cores escolhido para este desenvolvimento foi o YCrCb, pelos motivos já apresentados no referencial teórico.

Tendo todas as imagens do banco de dados representados neste espaço de cores, criou-se um histograma CrCb para cada grupo de imagens. Chamar-se-ão esses histogramas de modelos de cores.

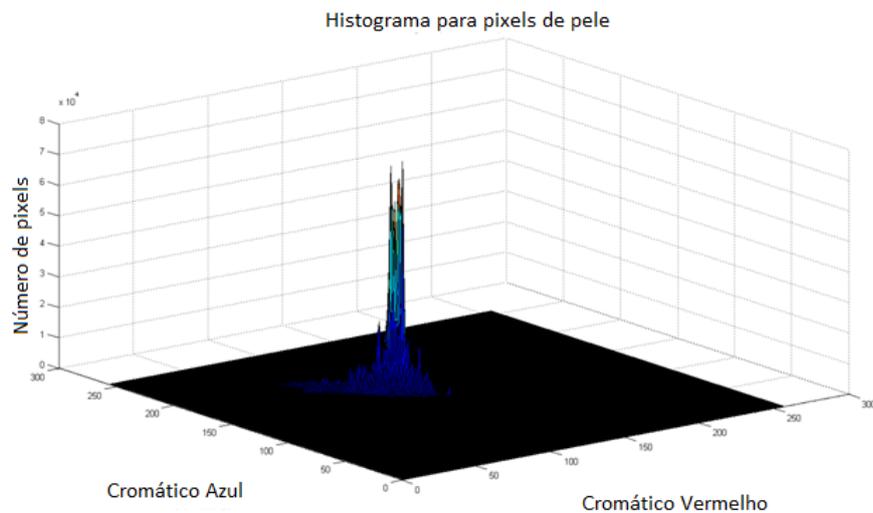


Figura 18 – Histograma das cores de pele.

Fonte: Autoria própria.

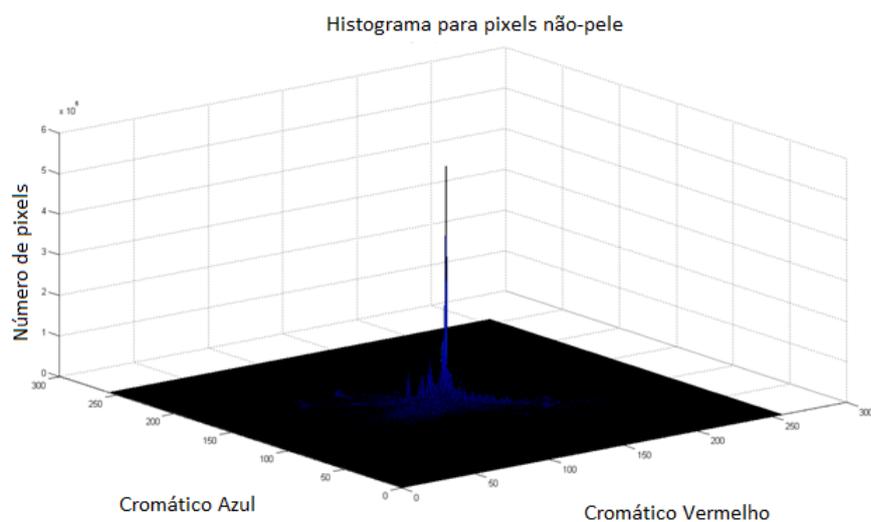


Figura 19 – Histograma das cores que não são de pele.

Fonte: Autoria própria.

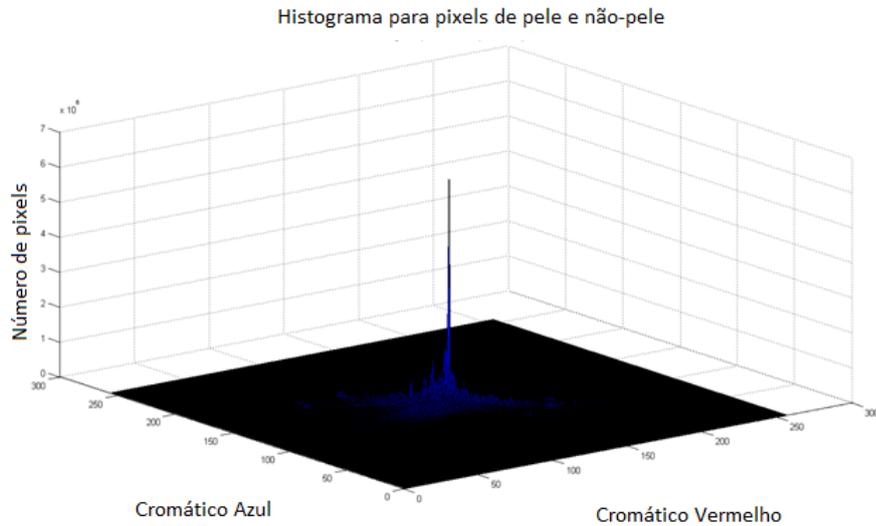


Figura 20 – Histograma das cores em geral.

Fonte: Autoria própria.

Em posse dos histogramas, é possível fazer um pequeno estudo estatístico da porcentagem das cores presentes em pixels correspondentes à pele em relação ao conjunto de todas as cores existentes. Utilizou-se a técnica de histograma com um classificador Bayesiano para fazer a classificação dos pixels em uma imagem (PHUNG; BOUZERDOUM; SR, 2005).

Ao sabermos a probabilidade condicional da cor de um pixel no espaço YCrCb, considerando que ele é um pixel com cor correspondente à pele:

$$P(CrCb^* | pele) = \frac{[CrCb^*]_{pele}}{Total_{pele}} \quad (2.1)$$

E a probabilidade condicional da cor de um pixel no espaço YCbCr, considerando que ele é um pixel com cor não correspondente a pele:

$$P(CrCb^* | npele) = \frac{[CrCb^*]_{npele}}{Total_{npele}} \quad (2.2)$$

Podemos utilizar um classificador Bayesiano, com a regra de decisão para o mínimo custo (FUKUNAGA, 1990):

$$\frac{P(CrCb^* | npele)}{P(CrCb^* | pele)} \leq \alpha \quad (2.3)$$

Onde:

- $[CrCb^*]_{pele}$ é o número de pixels com a cor CrCb* no histograma de pele;
- $[CrCb^*]_{npele}$ é o número de pixels com a cor CrCb* no histograma de não pele;

- $Total_{pele}$ é o número total de pixels contados no histograma de pele;
- $Total_{npele}$ é o número total de pixels contados no histograma de não pele;
- α é um valor de *threshold*

Se essa razão for menor ou igual que α , considera-se o pixel como pixel de pele. Caso contrário, é um pixel não pele. A lógica desta equação está no seguinte raciocínio: Se uma determinada cor tem uma probabilidade muito maior de aparecer no domínio de cores de não pele do que no domínio de cores de pele, ao encontrar esta cor em uma imagem, a probabilidade dela ser de não pele é muito maior.

Não existe nenhuma regra que defina o valor de α . Ele é diretamente relacionado às imagens utilizadas em cada uma das classes. Desta forma, determinou-se o valor de α empiricamente, testando o segmentador de pele em diversas áreas de pele e verificando a porcentagem de positivos, falsos positivos, negativos e falsos negativos.

É importante salientar que a eficiência deste método depende da qualidade das amostras do banco de dados: Cada grupo de imagens deve conter pixels que correspondam somente a sua classe.

Este método de segmentação mostrou uma boa precisão nas detecções de pixels de pele e é extremamente rápido pois as probabilidades são calculadas de valores já guardados em memória, agindo basicamente como uma máscara. A seguir, uma imagem mostrando a segmentação da pele:

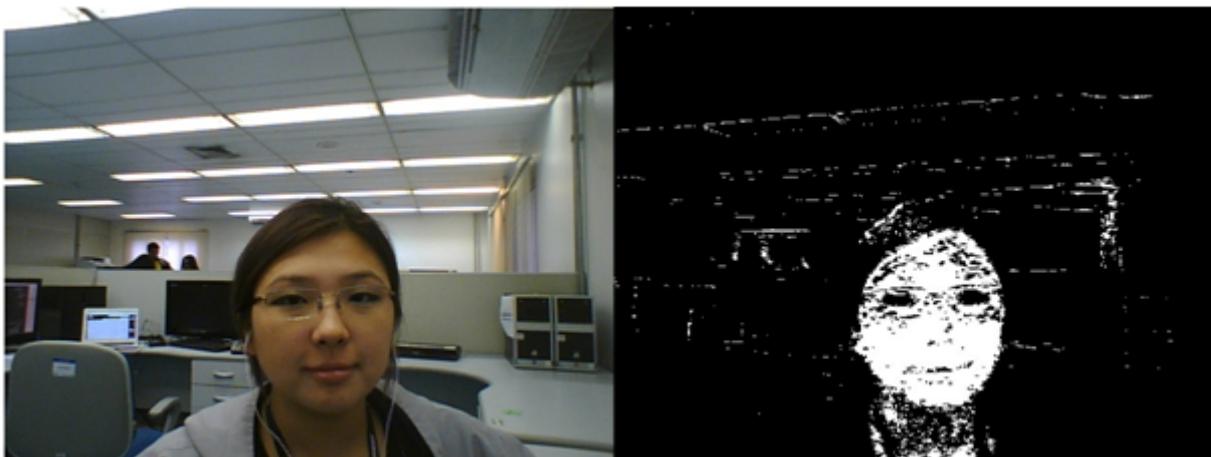


Figura 21 – Segmentação da pele.

Fonte: Autoria própria.

2.2.2 Detecção dos Elementos do Rosto

Com o segmentador de faces implementado, passou-se para o próximo passo, a verificação da presença de olhos e da boca. A biblioteca de visão computacional OpenCV possui uma classe especificamente criada para lidar classificadores Haar em cascada, chamada

CascadeClassifier. O OpenCV também já possui classificadores treinados para identificar olhos e bocas. Inicialmente, tentou-se treinar novos classificadores para encontrar essas características, mas devido ao fato do treinamento ser extremamente demorado e difícil (a escolha das imagens para treinamento precisa ser feita de forma muito precisa), decidiu-se utilizar os classificadores já disponíveis.

Para identificar se o segmento de pele encontrado no passo anterior tem um rosto ou não, verifica-se a presença dos dois olhos e da boca. Além disso, também se verifica algumas restrições geométricas como:

- Se a coordenada x do olho direito é menor que a coordenada x do olho esquerdo;
- Se as coordenadas y dos olhos é menor que a coordenada y do nariz;
- Se a coordenada y do nariz é menor que a coordenada y da boca;

Essas restrições partem do pressuposto de que o rosto a ser detectado nunca aparecerá de cabeça para baixo ou com uma inclinação muito alta.

Se estas restrições são respeitadas, considera-se a região como um rosto.



Figura 22 – Rosto Detectado.

Se observarmos a imagem de segmentação de pele, percebe-se que há diversas áreas muito pequenas de pixels considerados de pele e que não tem como realmente representar o rosto. Baseado nisso, pode-se aumentar ainda mais a eficiência do processo descartando essas pequenas áreas e efetuando a verificação apenas em áreas maiores do que um determinado tamanho.

Como já mencionado, a detecção de face retorna um objeto face e este é necessário para o reconhecimento facial.

2.3 Processamento do Rosto

Em geral, os algoritmos de classificação e reconhecimento possuem um bom desempenho quando as amostras de treinamento e as amostras de teste são obtidas em ambientes controlados. Por ambientes controlados quer se dizer mesma distância entre a pessoa ou objeto e a câmera, mesmo alinhamento do objeto com a câmera e, principalmente, mesmas condições de iluminação. Entretanto, em ambientes não controlados, os resultados caem drasticamente e isso se deve ao fato da grande diferença que a iluminação pode causar em um rosto (TAN; TRIGGS, 2010). Apesar deste projeto pressupor um ambiente com iluminação controlada, a posição da pessoa diante da câmera pode variar e assim a iluminação no rosto da pessoa também pode acontecer. Desta forma, antes de se realizar o reconhecimento facial, é preciso eliminar ao máximo a influência dessas variáveis. Sendo assim, todos os rostos encontrados, sejam para o treinamento do Reconhecedor, quanto para o reconhecimento do mesmo, são processados de modo a tentar normalizá-los em relação a iluminação, rotação e escala.



Figura 23 – Fluxograma 2 - Processamento do Rosto.

Fonte: Autoria própria.

2.3.1 Normalização da Rotação

Para as rotações do rosto, em relação a câmera, como não se tem nenhum tipo de medidor de profundidade, muito menos não está sendo realizado tipo algum de homografia para estimação da distância entre a câmera e os diversos pontos do rosto, só é possível normalizar a rotação em relação ao plano (x,y) da imagem, ao alinharmos a linha que liga os dois olhos com o eixo x da imagem. Uma vez que o detector de rostos retorna um objeto face que possui o retângulo correspondente ao rosto e as posições dos olhos, nariz e boca, é possível fazer essa normalização.

Para o algoritmo, consideramos o olho direito como centro de rotação da imagem.

Para saber o ângulo de inclinação da imagem, encontra-se a direção de um vetor que liga o centro dos dois olhos, partindo do olho direito.

$$\vec{v} = c_{O.E} - c_{O.D} \quad (2.4)$$

Onde:

- \vec{v} é o vetor que tem a mesma direção que a reta que liga os dois olhos;
- $c_{O.E}$ é o ponto central do olho esquerdo;
- $c_{O.D}$ é o ponto central do olho direito.



Figura 24 – Encontrando o vetor que tem a mesma direção que a reta que liga os dois olhos.

Após isso, encontra-se o ângulo θ do vetor, em relação ao eixo x da imagem, através das projeções ortogonais do mesmo.

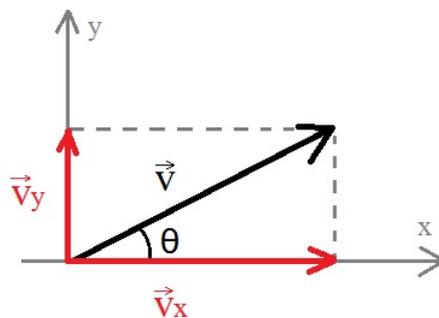


Figura 25 – Cálculo do ângulo theta.

O ângulo θ será então:

$$\theta = \arctan \left(\frac{|\vec{v}_y|}{|\vec{v}_x|} \right) \quad (2.5)$$

Sabendo-se o ângulo, pode-se definir a matriz de rotação que consiga mapear os pontos da imagem original para uma nova imagem rotacionada.

$$R = \begin{pmatrix} \cos \alpha & \sin \alpha & (1 - \cos \alpha) c_x - (\sin \alpha) c_y \\ -\sin \alpha & \cos \alpha & (\sin \alpha) c_x + (1 - \cos \alpha) c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

Onde $\alpha = -\theta$.

E após a rotação, recorta-se a imagem de forma a termos apenas o rosto.

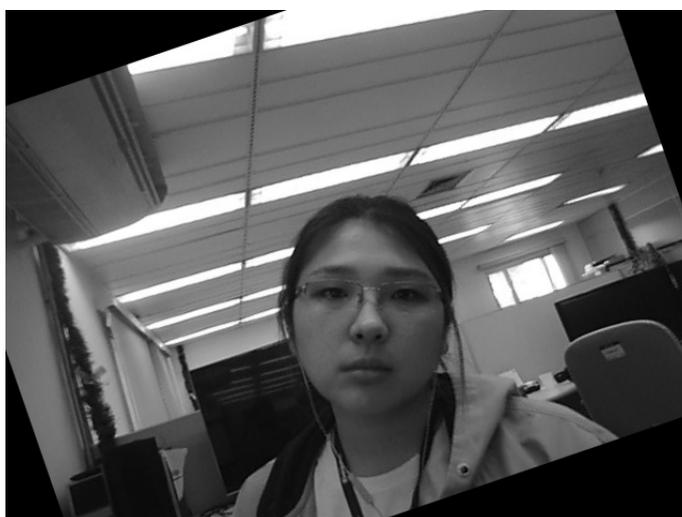


Figura 26 – Imagem rotacionada.

Fonte: Autoria própria.



Figura 27 – Imagem recortada.

Fonte: Autoria própria.

2.3.2 Normalização da Iluminação

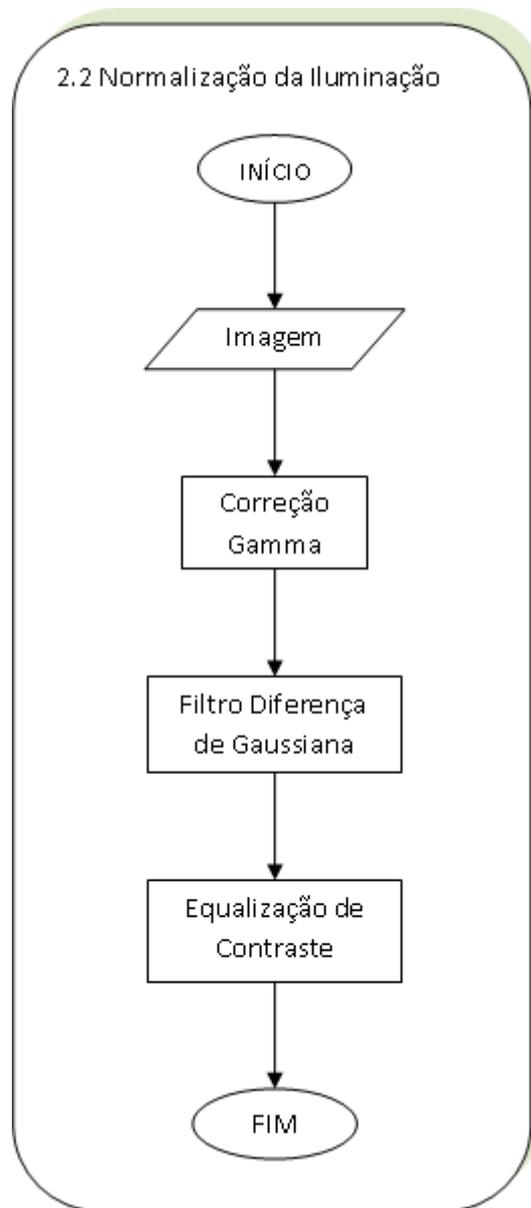


Figura 28 – Fluxograma 2.2 - Normalização da Iluminação.

Fonte: Autoria própria.

Para normalizar a iluminação nas imagens, utilizou-se o método proposto por (TAN; TRIGGS, 2010). Este método consiste de três principais passos: Correção Gamma, Filtro por Diferença de Gaussiana e Equalização do Contraste. A imagem a ser normalizada precisa estar na escala de cinza.

2.3.2.1 Correção Gamma

A correção Gamma é uma transformação linear de níveis de cinza, que substitui cada pixel com intensidade I na imagem por I^γ , com $0 \leq \gamma \leq 1$ ou $\log(I)$, caso $\gamma = 0$. Desta

forma, a correção gamma aumenta a faixa dinâmica de uma imagem.

Considerando uma resultado desse processo, uma imagem com valores de intensidade que vão de 0 a 255:

- Se $\gamma = 1$, tem-se uma transformação indentidade, onde cada pixel é mapeado para ele mesmo;
- Se $\gamma = 0,5$, observa-se que os pixels de valores de 0 a 50 são mapeados a uma faixa de valores de 0 a 115, enquanto pixels que vão de 200 a 255 são mapeados para uma faixa menor que vai de 230 a 255;

Em outras palavras, as áreas escuras da imagem se tornam mais claras e as mais claras praticamente mantêm sua intensidade. Desta forma, aumenta-se a visibilidade dos contrastes. O método sugere utilizar um valor de γ entre 0 e 0,5, valores que proveem uma significativa melhora no contraste da imagem, sem amplificar demasiadamente os ruídos. Para este trabalho, utilizou-se o valor padrão sugerido pelo método de 0,2.



Figura 29 – Correção Gamma.

Fonte: Autoria própria.

2.3.2.2 Filtro Diferença de Gaussiana

A correção Gamma ou qualquer outro algoritmo de normalização de contraste não remove os efeitos dos gradientes de intensidade, tais como efeitos da sombra. Esses efeitos são predominantemente um efeito de baixa frequência.

Entretanto, não é possível diferenciar um gradiente de iluminação causado por sombras e um causado devido a estrutura do objeto, uma vez que a iluminação também é modelada como um fenômeno de baixa frequência. Para se remover esses efeitos, pode-se utilizar um filtro passa-alta.

Por outro lado, ruídos randômicos na imagem são fenômenos de alta frequência e, desta forma, se mostra interessante o uso de um filtro passa-banda. Uma forma de se fazer um filtro passa-banda é utilizar a Diferença de Gaussianas.

Ele é considerado uma aproximação do Laplaciano da Gaussiana, técnica usada para detecção de bordas. Como o próprio nome diz, este filtro é apenas a subtração de duas gaussianas com diferentes valores de sigma.

Ao aumentar o sigma da gaussiana menor, consegue-se bordas mais grossas. Desta forma é interessante não aumentá-lo muito para não perder detalhes finos da imagem. Um sigma pequeno é mais interessante pois ele irá eliminar apenas o ruído.

A segunda gaussiana possui um sigma maior, que irá remover os detalhes de alta frequência na imagem, deixando só os componentes de baixa frequência na imagem.

Ao se subtrair esta gaussiana de baixa frequência da imagem que foi filtrada com um passa-alta, obtém-se uma imagem de bordas.

Em geral, uma relação de 1:2 entre os dois valores de sigma provê bons resultados. Esses valores também foram utilizados conforme recomendados pelo método:



Figura 30 – Filtro Diferença de Gaussiana.

Fonte: Autoria própria.

2.3.2.3 Equalização de Contraste

Este é o último passo do pré-processamento da imagem. Executa-se uma equalização de contraste que redimensiona globalmente as intensidades da imagem para uma medida padrão robusta do contraste em geral ou da variação da intensidade.

Uma vez que se executou uma diferença de Gaussianas no passo anterior, é bem provável que valores extremos sejam produzidos por sombras e ruído. Poderia, por exemplo, usar a mediana do valor absoluto do sinal, mas uma forma mais computacionalmente barata é quebrar esse processo em dois estágios:

$$I(x, y) \leftarrow \frac{I(x, y)}{(\text{mean}(|I(x', y')|^\alpha))^{\frac{1}{\alpha}}} \quad (2.7)$$

$$I(x, y) \leftarrow \frac{I(x, y)}{(\text{mean}(\min(\tau, |I(x', y')|)^\alpha))^{\frac{1}{\alpha}}} \quad (2.8)$$

Onde:

- α é um expoente compressor que reduz a influência de valores altos;
- τ um *threshold* usado para truncar valores alto depois da primeira fase da normalização e a média é em relação a imagem inteira.

Após estas duas operações, a imagem já está bem escalonada, mas ainda pode conter valores extremos. Aplica-se, então, uma função não-linear para comprimir esses valores, através de uma função tangente hiperbólica:

$$I(x, y) \leftarrow \tau * \tanh\left(\frac{I(x, y)}{\tau}\right) \quad (2.9)$$

O resultado desse processo é uma image cujos valores variam na faixa de $(-\tau, \tau)$. Para conseguir o resultado final, só é preciso normalizar a imagem para valores entre 0 e 255.



Figura 31 – Contrast Equalization.

Fonte: Autoria própria.

Ao eliminar-se a influência da luz, a extração de features das características das imagens se torna mais eficiente e precisa pois as características a serem retiradas das imagens serão provenientes das imagens em si e não de efeitos introduzidos nelas pelos distúrbios.

2.3.3 Normalização da Escala

Mesmo que se estabeleça uma distância entre o indivíduo e a câmera, dificilmente o rosto a ser detectado terá sempre o mesmo tamanho. Para a implementação do reconhecedor através da utilização das Máquinas de Vetores de Suporte, é necessário que todas as imagens possuam o mesmo número de *features*. Considerando estes como os Padrões Binários Locais, que é calculado para cada pixel na imagem, é preciso que as imagens possuam mesmo tamanho. Uma forma simples de se fazer isso é redimensionando todas as imagens para um mesmo tamanho.

2.4 Reconhecedor de Rostos

Para a implementação do reconhecimento, levou-se em consideração dois pontos:

1. O reconhecedor deve ser capaz de distinguir se a pessoa faz parte ou não da base de dados;
2. O reconhecedor deve distinguir corretamente entre as pessoas dentro da base de dados;

Devido à grande complexidade que rostos apresentam, a utilização de Máquinas de Vetores de Suporte é interessante pois eles evitam inúmeras dificuldades de se utilizar funções lineares em espaços de *features* de alta dimensionalidade. A abordagem convencional utilizando SVM para classificação é eficiente para o segundo item, mas não para o primeiro. Com o hiperplano encontrado, ele modela o espaço inteiro de rostos em apenas duas classes. Desta forma, ao se deparar com um rosto fora da base de dados, ele consegue dizer apenas em qual dos lados do hiperplano este rosto se encontra. Ou seja, se aparecer alguma pessoa parecida com a registrada na base de dados, ela será reconhecida como esta pessoa.

O SVM, em sua natureza, trata apenas de problemas binários. No caso da aplicação do projeto, onde mais de uma pessoa pode estar cadastrada na base de dados, tem-se um problema de multi-classe. Existem trabalhos que estenderam o SVM para um problema de multi-classe como em (WESTON; WATKINS, 1998), (CRAMMER; SINGER, 2002), (LEE; LIN; WAHBA, 2004), onde se tenta encontrar uma função de decisão levando em consideração todas as classes de uma só vez.

Entretanto, de forma a manter a simplicidade do SVM, foram criadas duas abordagens multi-classe utilizando SVM binários, sendo uma das mais utilizadas a *One versus All*, ou Um contra Todos (RIFKIN; KLAUTAU, 2004). Esta abordagem define o treinamento de um SVM para cada classe, onde as amostras da classe são definidas como as amostras positivas e todas as amostras das outras classes são negativas. Para se determinar a classe de uma amostra de teste, verifica-se qual dos SVM retornou a *label* positiva.

Entretanto, nenhuma delas define um parâmetro para determinar se uma amostra pertence a uma classe conhecida ou se ela é desconhecida. (EKENEL; SZASZ-TOTH; STIEFELHAGEN, 2009) propõe em sua modelagem do problema de *openset* a utilização da distância entre a amostra e o hiperplano definido pelo SVM:

$$d(x_i) = K(\omega, x_i) + b \quad (2.10)$$

É certo que quanto maior for esta distância mais perto de uma das classes esta amostra vai estar. Entretanto, existe o problema de definir quão próxima uma amostra tem que estar da classe para ser considerada parte dela, uma vez que o SVM define o hiperplano que maximiza a distância entre ele e as amostras de treinamento.

Pensemos de uma maneira diferente, onde ao invés de tentar separar as amostras neste espaço de maior dimensionalidade, tentamos encontrar uma função de rostos.

Imaginemos que este espaço de maior dimensionalidade é um espaço onde se pode mapear o conjunto de *features* de um rosto a sua classel. Idealmente, haveria um conjunto para cada classe. Entretanto, o rosto é passível de diversas distorções globais (posição em relação a câmera, rotação) e locais (formato da boca muda de acordo com a expressão, assim como as posições das sobrancelhas etc). Mas sendo as variadas amostras pertencen-

tes a mesma pessoa, elas devem ser mapeadas nesta função de rosto próximas umas das outras.

Este tipo de regressão é possível através do SVR. Por ser uma regressão, o SVR permite diversas classes, que indicam o valor da amostra no espaço de maior dimensionalidade. Entretanto, para o problema deste projeto, vamos considerar apenas dois valores: 1 caso a amostra for positiva e 0, caso seja negativa. Utilizando-se a abordagem Um Contra Todos, conforme o número de classes cresce, aumenta-se também o conhecimento sobre a classe negativa. Só é preciso tomar cuidado pois o número de amostras negativas também cresce e isso pode acabar desbalanceando a base de treinamento, prejudicando a performance do SVM. Desta forma, para este trabalho, optou-se por modelar as classes negativas dos SVR com:

$$n_{amostras} = \frac{\text{tamanho}_{classe}}{2 * (n - 1)} \text{por classe} \quad (2.11)$$

Assim, o número de amostras positivas e amostras negativas será sempre o mesmo, balanceado a base de dados de treinamento. Entretanto, construir a base de dados desta forma restringe o número de classes para, no máximo, $\frac{\text{tamanho}_{classe}}{2} + 1$.

Se tentarmos encontrar a posição de amostras de testes através destas funções de rostos definidas pelo SVR, perceberemos que elas retornarão valores mais próximos de 1, caso a amostra seja mais parecida com a classe positiva e valores mais próximos de 0, caso seja mais parecida com a classe negativa. Para treinar o SVR, utilizou-se a técnica de *cross-validation* juntamente com uma busca *grid*. Esta técnica testa diversos valores para os parâmetros do SVR (no caso, os valores de C e de ϵ), subdividindo as amostras de treinamento em vários grupos, treinando o SVR com um destes conjuntos e testando o modelo contra os outros grupos até encontrar o menor erro de previsão.

Neste momento, nos deparamos com o mesmo problema do SVM: Como definir o quanto uma amostra deve ser próxima ao valor de 1 para que seja considerada parte da classe positiva?

Para definir isto, realizou-se o treinamento de um SVR, tomando como amostras positivas pertencentes a uma pessoa e amostras negativas pertencentes a outra dentro da base de dados. Como *features* utilizou-se os padrões binários locais. Após isso, verificou-se os valores retornados pelo SVR na regressão das amostras de teste. Observou-se que para amostras pertencentes a classe positiva, estes valores estavam em sua maioria acima da média dos valores retornados pelo SVR na regressão das amostras positivas menos um desvio padrão destes mesmos valores.

Valores retornados por um SVR		
n	Label da Amostra	Valor
0	1	0,808
1	1	0,811
2	1	0,787
3	1	0,775
4	1	0,686
5	1	0,926

Tabela 1 – Valores retornados pelo SVR.

Para este teste, a média foi de 0,889, com desvio padrão de 0,085. Foram analisadas 90 imagens, sendo delas 45 amostras positivas e 45 negativas. Das amostras positivas, 38 retornaram um valor acima da média menos o desvio padrão. A definição deste valor foi feita de forma empírica:

$$ValorLimiar = média_{valores+} - desvioPadrão_{valores+} \quad (2.12)$$

Desta forma, definiu-se o treinamento do reconhecedor da seguinte maneira:

1. Obter as amostras dos indivíduos a serem cadastrados na base de dados do reconhecedor;
2. Dividí-los em amostras de treinamento e de teste;
3. Treinar os SVRs seguindo a abordagem Um Contra Todos;
4. Verificar os valores retornados pelos SVRs em cima das amostras de teste e calcular o valor limiar de cada classe;

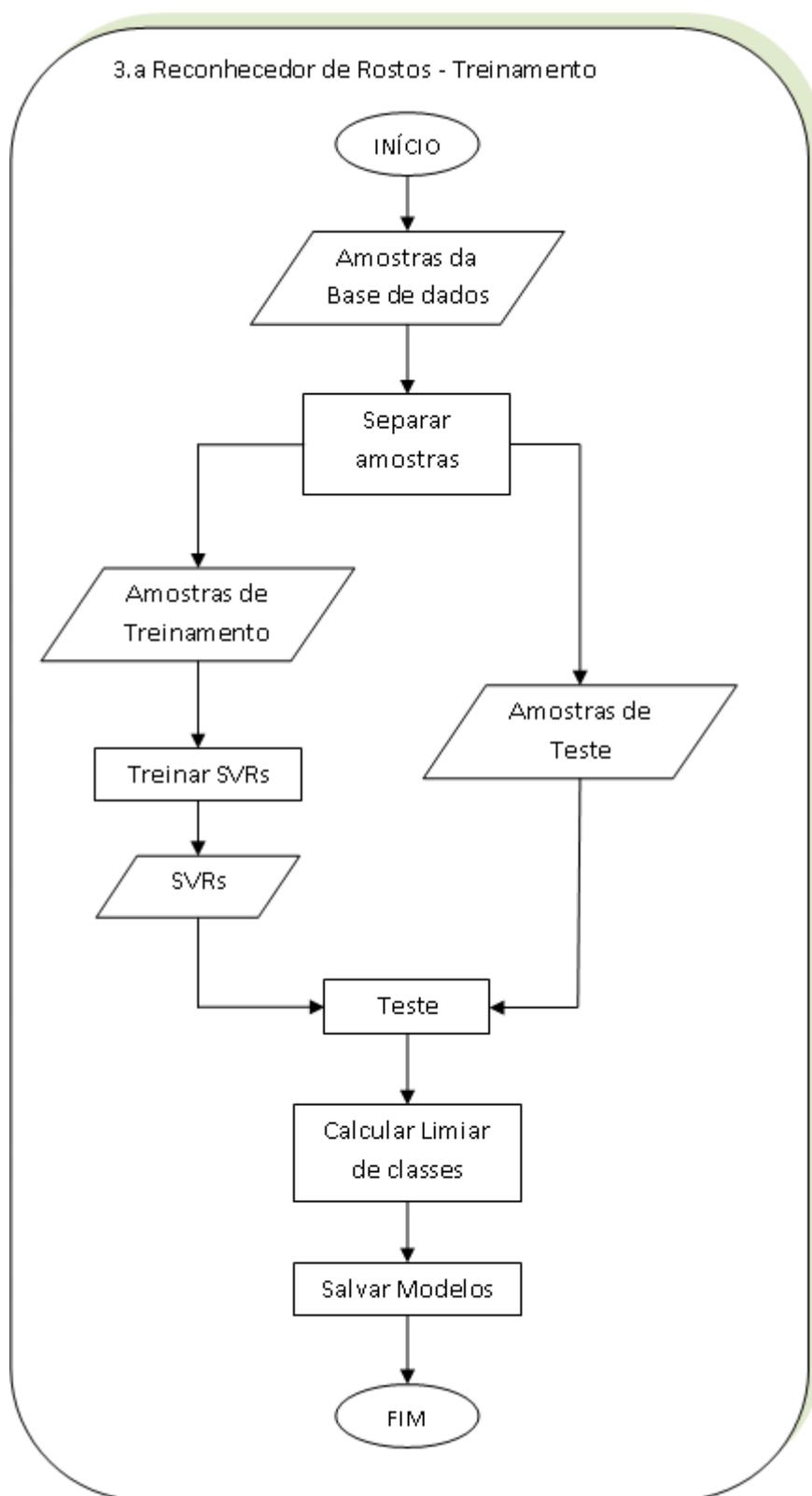


Figura 32 – Fluxograma 3.a - Treinamento do Reconhecedor.

Fonte: Autoria própria.

O reconhecimento ficou então definido:

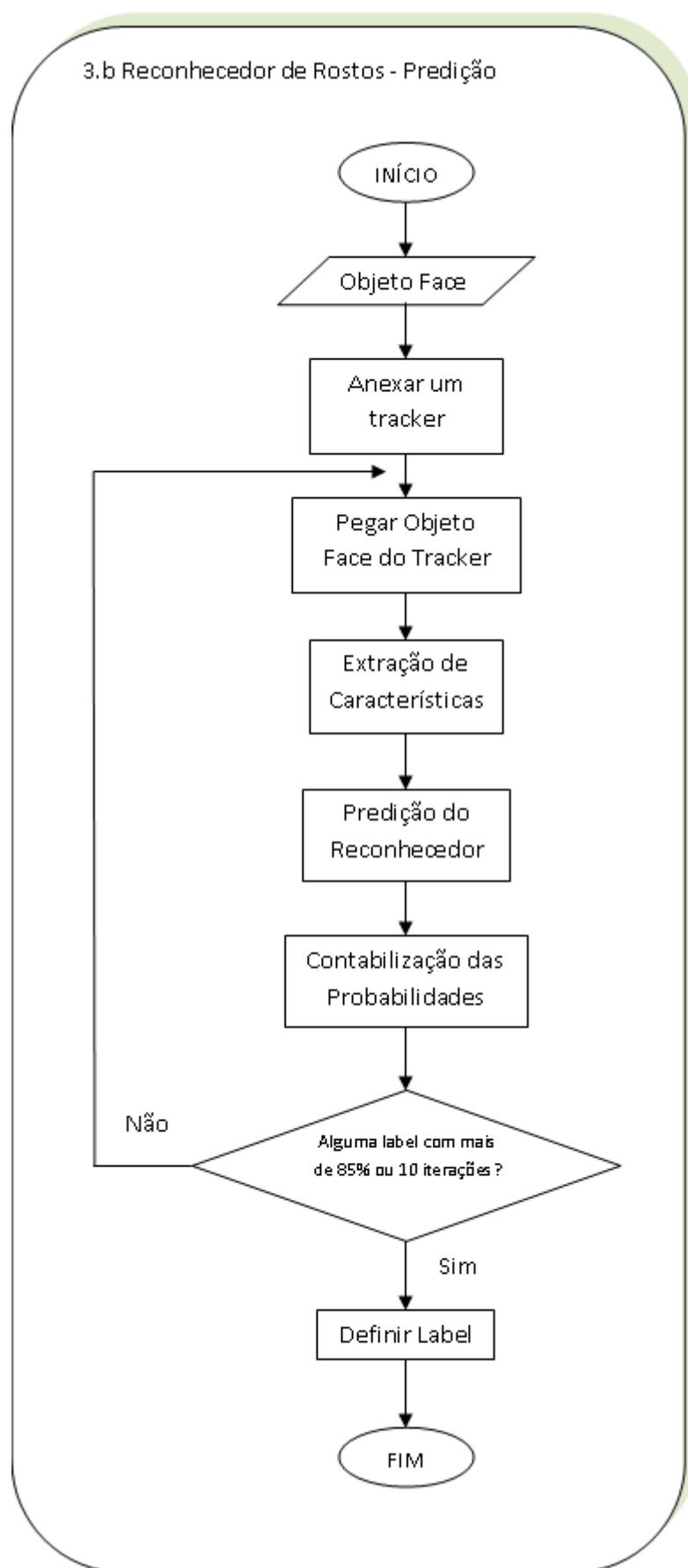


Figura 33 – Fluxograma 3.b - Reconhecimento.

Fonte: Autoria própria.

1. Fazer a regressão da amostra de teste por todos os SVRs;
2. Os SVRs que retornarem valores acima de limiar da sua respectiva classe positiva, darão um voto para esta classe;
3. No final, contabiliza-se o total de votos. Se nenhuma classe tiver voto, a amostra é de uma pessoa desconhecida. Se mais de uma classe tiver votos, a classe que tiver sido votada pelo SVR com valor de regressão mais próximo de 1 é escolhida.

Durante o reconhecimento, as amostras de testes são obtidas de um *stream* de vídeo. Dependendo da posição da pessoa diante a câmera durante a captura, é possível que o reconhecedor se confunda. De forma a aumentar ainda mais a taxa de reconhecimento, pega-se uma sequência de 5 ou mais quadros e verifica-se a identidade do rosto nestes. No primeiro frame, o reconhecedor pode retornar classe 1, mas no segundo, retornar classe 2. Conforme se verifica as identidades na sequência, checa-se qual classe foi retornada mais vezes. E quando uma das classes recebe mais que 85% dos votos, ela é a escolhida como identidade da pessoa. Também foi imposta uma condição de parada, pois é possível que este algoritmo nunca convirja para um valor, retornando como desconhecido depois de 10 iterações.

Entretanto, para se fazer esta verificação, é preciso ter certeza de que o mesmo rosto está sendo verificado em cada quadro. É possível que entre a captura de um quadro e outro, outra pessoa se coloque em frente a câmera. Para isso, a cada rosto encontrado, um rastreador de posição é anexado a ele e a verificação da identidade do rosto é realizada em cima dos objetos faces retornados pelo rastreador. Sendo assim o primeiro rosto a ser verificado em uma sequência é retornado pelo detector de faces, mas os seguintes são retornados pelo rastreador, até que a identidade do mesmo seja encontrada.

2.4.1 Rastreador de rostos

A implementação do rastreador de rostos seguiu a explicação dada por (BAGGIO, 2012). Ele define dois modelos para que o rastreamento seja possível: O *shape model* e o *patch model*.

O *shape model* é um modelo que descreve a geometria da face humana. Para tanto, utilizou-se as 3755 imagens manualmente pré-anotadas da base de dados da MUCT (MILBORROW; MORTEL; NICOLLS, 2010) cada uma com 76 pontos marcados. Estabeleceu-se restrições geométricas e pontos de simetria do rosto de modo a criar um modelo linear que consiga fazer uma representação paramétrica compacta de como o formato do rosto pode variar com vários níveis de complexidade. A principal ideia da modelagem linear das formas do rosto é conseguir representar cada conjunto de N pontos de um rosto em um plano de 2N dimensões, onde todos os pontos do rosto se encontram.

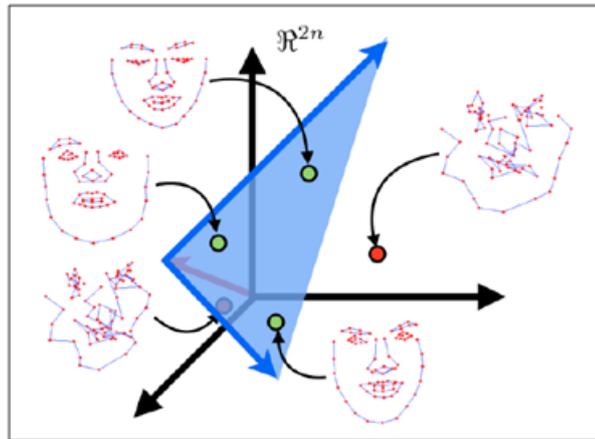


Figura 34 – *Shape Model* - Subespaço representado pelo modelo linear.

Fonte: (BAGGIO, 2012)

Já o *patch model* é um modelo constituído de diversas imagens lineares que representam a aparência de cada ponto marcado nos rostos da database. É fazendo a correlação entre a área deste ponto no rosto a ser rastreado com *patch* deste ponto no modelo que se obtém o rastreamento.

Para que o rastreamento seja bem sucedido, em primeiro lugar, é preciso encontrar o rosto e a partir da localização dada pelo detector de rosto, marcam-se os pontos no frame de acordo com a similaridade entre o *patch model* e o rosto no frame. O *shape model* assegura que, mesmo que um ponto se assemelhe mais a uma parte da imagem, ele não será marcado lá, caso viole as regras de restrição geométrica estabelecidas por ele.



Figura 35 – Rastreador de Rosto.

Fonte: Autoria própria.

3 Resultados Obtidos

3.1 Segmentador de Pele

Para a implementação do detector de rostos, foi necessária a execução de diversos testes para a definição do parâmetro α do segmentador de pele. Variando-se este parâmetro, o segmentador se torna mais ou menos tendencioso a classificar a cor de um pixel como cor de pele.

Para tanto, criou-se uma base de dados de teste, seguindo as mesmas diretrizes da criação da base de treinamento, em cujas imagens aplicou-se o segmentador de pele com variados valores de α . Calculando-se as porcentagens de precisão, sensibilidade, especificidade e acurácia, foi possível a construção do seguinte gráfico ROC:

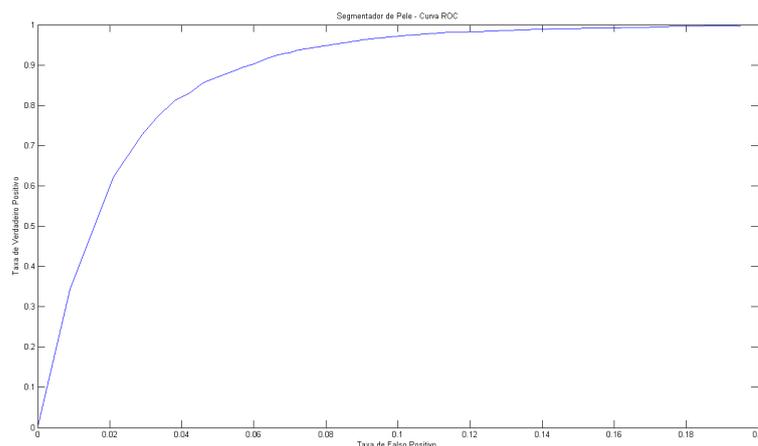


Figura 36 – Curva ROC para o Segmentador de Pele.

Variou-se α de 0 a 30. O ponto (0,0) do gráfico foi obtido com $\alpha = 0$. Neste ponto, independente da distribuição probabilística retornada pelos histogramas de cores, sempre se terá um valor maior ou igual a zero e desta forma, todos os pixels serão classificados como pixels de não pele. Conforme se aumenta este valor de α , começa-se a ter uma melhor sensibilidade aos pixels de cor de pele e o ponto ótimo para este modelo, ou seja, o mais próximo do ponto (1,0) do gráfico ocorre quando $\alpha = 0,4$.

Após a definição do valor de α , testou-se este segmentador em 500 imagens da base de dados LFW(HUANG et al., 2007), obtendo-se os seguintes valores:

Segmentador de Pele - $\alpha = 0,4$	
Parâmetro	Porcentagem
Precisão	98,44
Sensibilidade	82,91
Especificidade	95,77
Acurácia	95,61

Tabela 2 – Segmentador de Pele ($\alpha=0,4$).

Pelos valores obtidos, percebe-se que, devido à alta porcentagem de especificidade, o segmentador consegue identificar bem cores que não são de pele. Assim como a alta porcentagem de sensibilidade indica que o segmentador consegue identificar bem cores que são de pele.

A razão pela ocorrência de falsos positivos se encontra no fato de que muitas das cores consideradas de pele também podem representar outras superfícies. Pode-se notar na seguinte imagem a confusão do classificador ao se deparar com tons amarelados e alaranjados:



Figura 37 – Segmentador de Pele - Ocorrência de falso positivo.

Fonte da imagem original:(MORESCHI, 2011)

Ocorrem também os falsos negativos, onde o classificador falha na detecção das cores de pele. Apesar de termos utilizado o espaço de cores YCrCb, ele não é totalmente invariante às mudanças de iluminação e em algumas situações, a pele não é encontrada:

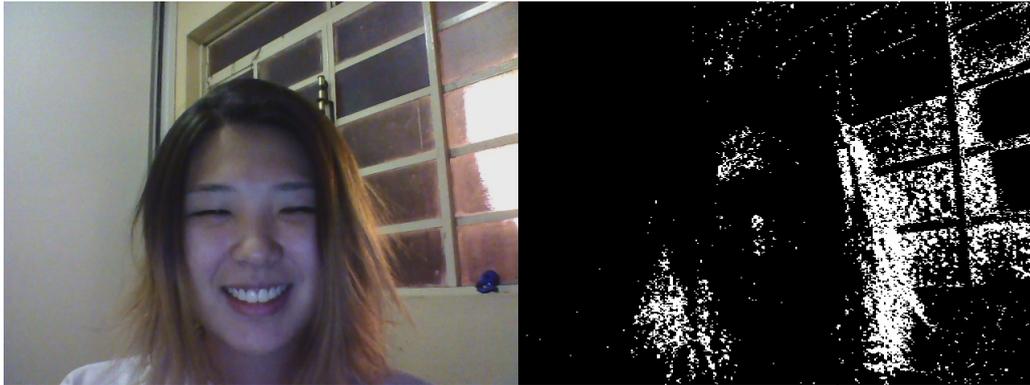


Figura 38 – Segmentador de Pele - Ocorrência de falso negativo.

Fonte da imagem original: (HUANG et al., 2007)

Mas mesmo com estes erros, com base na curva ROC e nas medidas de Sensibilidade e Especificidade, conclui-se que o segmentador obtido é capaz de realizar uma boa segmentação da pele.

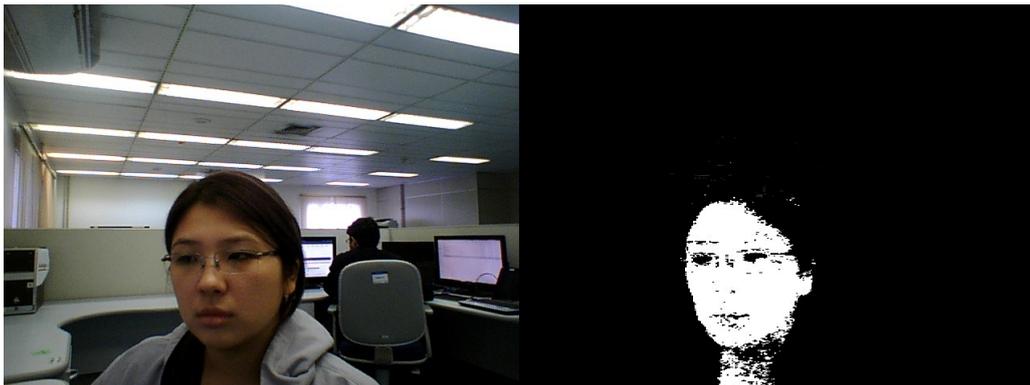


Figura 39 – Segmentador de Pele - Imagens corretamente segmentadas.

Fonte da imagem original: (HUANG et al., 2007)

3.2 Detector de Rosto

Para verificar o desempenho do detector de rosto proposto por este trabalho, foram realizados dois testes: Um para verificar capacidade de detecção de rosto e outro para verificar a velocidade do método.

Para o primeiro teste, foram utilizados outros 200 rostos da base de dados LFW mais 95 imagens sem rosto algum, retiradas da *Internet*, verificando as taxas de precisão, sensibilidade, especificidade e acurácia do mesmo. Para fins de comparação, fez-se o mesmo teste utilizando apenas os classificadores Haar em cascata.

Foi considerado como positivos os rostos corretamente localizados, juntamente com os elementos de rosto também corretamente encontrados. Os rostos com elementos encontra-

dos em locais errados ou em imagens sem rosto foram considerados como falsos positivos. Amostras com rosto cuja detecção retornou negativa foi considerado como falsos negativos.

Os resultados se encontram a seguir:

Detector de Rosto - Haar em Cascata	
Parâmetro	Porcentagem
Precisão	93,66
Sensibilidade	69,63
Especificidade	91,34
Acurácia	77,28

Tabela 3 – Detector de Rosto - Apenas com classificadores Haar em Cascata.

Detector de Rosto - Método Proposto	
Parâmetro	Porcentagem
Precisão	90.18
Sensibilidade	75.55
Especificidade	88.26
Acurácia	78.30

Tabela 4 – Detector de Rosto - Método proposto.

Comparando os resultados para ambos os detectores, percebe-se que o método proposto possui vantagens sobre a utilização de apenas os classificadores Haar em Cascata. O método proposto apresenta uma sensibilidade maior, o que significa que, das amostras positivas presentes na base de dados de teste, ele foi capaz de encontrar corretamente mais rostos que o método pelo Haar em Cascata. Isso acaba refletindo na acurácia do método, pois encontrando mais rostos corretamente, aumenta-se o número de detecções corretas (sejam elas positivas ou negativas).

Entretanto, o método pelo Haar em Cascata apresentou uma maior precisão e especificidade. Isso significa que das amostras identificadas como positivas, ele identificou corretamente mais rostos que o método proposto. Além disso, das amostras negativas presentes na base de dados de teste, ele foi capaz de detectar corretamente mais rostos (ou no caso, a ausência deles) que o método proposto.

Estas diferenças, entretanto, foram pequenas, mostrando que se conseguiu desenvolver um método de detecção facial com precisão e acurácia comparável ao método convencional.

O segundo teste foi realizado conjuntamente com o primeiro teste, medindo-se os tempos de detecção do rosto em sua totalidade e de detecção dos elementos. Foram contabilizados apenas os tempos de detecção de rostos corretamente encontrados. De todos

os tempos, verificou-se a média de 20 deles para que possamos ter uma ideia do tempo de detecção, em geral.

Encontrou-se os seguintes resultados:

Tempos (ms) de detecção pelo Classificador Haar em Cascata							
n	Rosto	O. D.	O. E.	Nariz	Boca	Elementos	Total
1	312,396	11,459	7,569	15,985	13,386	48,399	360,795
2	310,788	8,895	7,035	14,463	11,695	42,088	352,876
3	307,971	9,502	9,603	16,292	12,873	48,270	356,241
4	308,299	12,422	7,321	15,615	13,024	48,382	356,681
5	310,769	10,554	8,796	15,714	12,458	47,521	358,290
6	312,947	11,415	10,350	14,544	13,606	49,915	362,862
7	314,104	9,233	8,203	13,850	13,846	45,133	359,236
8	313,508	11,632	8,538	14,204	11,124	45,498	359,006
9	311,595	15,961	9,090	14,832	12,564	52,448	364,042
10	316,988	11,749	10,366	13,994	10,566	46,675	363,664
11	311,264	13,050	7,982	14,497	14,833	50,362	361,626
12	317,466	13,124	6,665	13,802	12,875	46,466	363,932
13	313,926	12,719	10,238	14,306	12,516	49,779	363,704
14	314,477	10,720	9,053	14,360	10,725	44,858	359,335
15	310,249	11,755	9,523	15,520	13,367	50,165	360,414
16	314,957	8,401	10,044	16,349	14,053	48,847	363,805
17	310,541	12,515	5,888	14,470	13,466	46,339	356,880
18	312,852	10,306	9,380	15,364	15,177	50,227	363,079
19	321,049	12,380	7,493	13,776	12,495	46,144	367,192
20	309,826	11,066	8,895	14,340	11,727	46,028	355,854
Média	312,799	11,443	8,602	14,814	12,819	47,677	360,476

Tabela 5 – Tempos (ms) de detecção pelo Classificador Haar em Cascata

Tempos (ms) de detecção pelo método proposto							
n	Segmentação	O. D.	O. E.	Nariz	Boca	Elementos	Total
0	4,614	12,732	7,668	26,758	10,981	58,140	62,754
1	2,920	13,642	8,305	32,835	9,132	63,914	66,834
2	4,351	11,668	9,491	21,626	13,195	55,980	60,331
3	3,496	13,888	8,834	27,797	12,049	62,568	66,063
4	2,886	11,786	7,452	31,322	8,223	58,782	61,668
5	3,988	8,458	6,729	23,599	9,261	48,048	52,036
7	2,818	12,963	9,315	30,580	11,611	64,469	67,287
9	2,905	14,157	8,960	24,892	11,205	59,214	62,119
10	3,454	11,152	10,899	22,736	13,745	58,531	61,986
11	4,146	11,545	9,521	29,733	12,624	63,422	67,568
12	3,455	13,816	7,637	35,179	13,109	69,742	73,196
13	4,715	12,737	11,276	29,426	15,480	68,919	73,634
14	4,140	18,740	9,699	30,160	11,867	70,466	74,606
15	4,729	17,758	10,511	54,264	15,385	97,918	102,648
16	3,302	17,742	12,252	41,182	17,271	88,447	91,749
17	4,440	18,733	12,463	37,144	11,105	79,444	83,884
18	3,220	11,115	8,607	34,781	10,454	64,957	68,177
19	3,185	13,650	9,551	27,218	10,866	61,284	64,469
20	3,271	12,526	7,140	25,199	11,641	56,505	59,776
Média	3,502	12,940	8,816	29,321	11,460	62,537	66,039

Tabela 6 – Tempos (ms) de detecção pelo método proposto

Percebe-se uma grande diferença entre o tempo total de cada detector. O método proposto detecta o rosto, juntamente com os elementos, em média, cinco vezes mais rápido que o método que utiliza apenas os classificadores Haar em cascata. Neste método, percebe-se que o que mais consome tempo é a detecção do rosto em si, pois os tempos de detecção dos elementos é basicamente o mesmo nos dois métodos, uma vez que ambos utilizam os mesmos meios de detecção para os elementos.

Quanto aos parâmetros de desempenho, percebeu-se que o maior problema de ambos os métodos está na detecção dos elementos de rosto. O classificador Haar em Cascata para olhos disponibilizado pela biblioteca livre *OpenCV* mostrou facilidade em encontrar rostos frontais ou quase frontais.

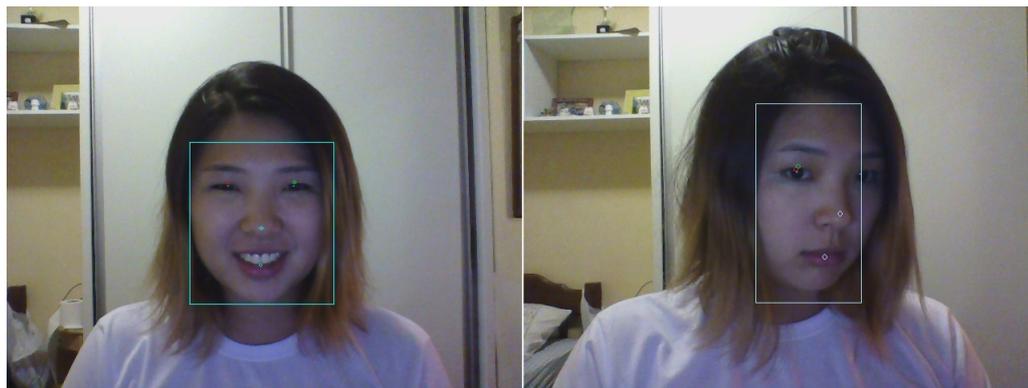


Figure 40 – Detecção pelo método proposto - a) Verdadeiro Positivo e b) Falso Positivo

Fonte das imagens originais: (HUANG et al., 2007)

Entretanto, em imagens mais laterais, é comum o detector encontrar os dois olhos no mesmo lado do rosto. Tentou-se definir áreas de interesse na busca de cada olho, mas isso resultou apenas na não detecção de um dos olhos. A imagem b) da figura 4.5 é um falso positivo pois apesar do detector ter retornado um objeto face, ele falhou na detecção dos olhos e isso acarretará em um erro no reconhecimento facial.

Outro ponto a se ressaltar é que ambos os detectores são falhos em casos onde a pessoa na amostra está utilizando óculos escuros. Os detectores também mostram dificuldade em encontrar rostos com óculos comuns cujas armações são muito robustas. Isso se deve ao fato de que os classificadores Haar em cascata utilizados para a detecção dos olhos não foi treinado para detectar olhos atrás de óculos.

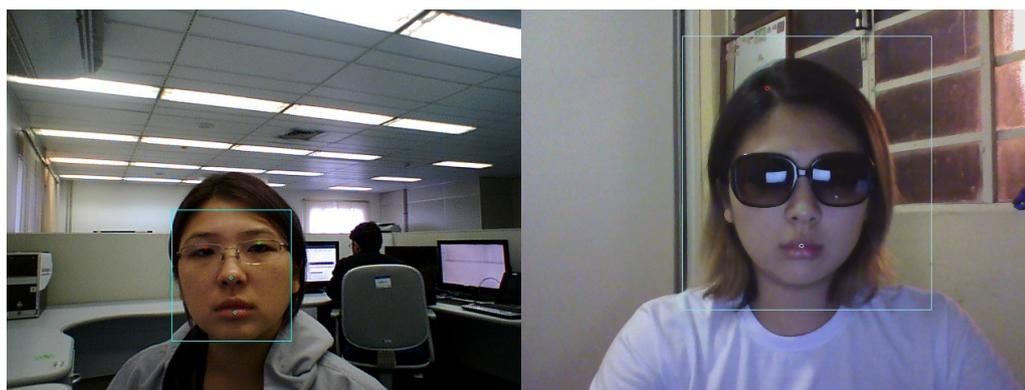


Figure 41 – Detecção pelo método proposto - Detecção em rostos com óculos.

Fonte das imagens originais: (HUANG et al., 2007)

Com base nestes dois testes, vê-se que foi possível o desenvolvimento de um novo método de detecção facial aplicado ao problema deste projeto, mais rápido do que o método convencional e sem afetar de forma significativa a taxa de detecção de rostos.

3.3 Reconhecedor de Rosto

Para a implementação do Reconhecedor de Rosto, foi necessário inicialmente determinar qual o melhor tamanho das amostras e qual o melhor número de amostras por classe deve-se utilizar no treinamento. Em seguida, para o tamanho e número de amostras selecionados, verificou-se a influência da utilização dos Padrões Binários Locais como *features*, assim como a combinação dos mesmos com a normalização da iluminação. Averiguou-se então qual o efeito sobre o desempenho do reconhecedor se construirmos a base de treinamento de acordo com a abordagem Um Contra Todos convencional e a abordagem proposta pelo trabalho. Por fim, verificou-se quanto a utilização de uma sequência de quadros melhora o desempenho do reconhecedor.

3.3.1 Tamanho e número de amostras para treinamento

A determinação desses dois parâmetros foi realizada em um teste conjunto. Treinou-se um SVR, combinando diversos tamanhos de amostras e diversos números de amostras por classe, verificando-se as taxas de precisão, sensibilidade, especificidade e acurácia. O tamanho das amostras foi variado de uma largura de 10 pixels a 100 pixels, com um acréscimo de 15 pixels a cada iteração, e altura sendo calculada como a *largura* * 1,2. O número de amostras variou de 60 a 120, com um acréscimo de 10 amostras a cada iteração. Também de grande importância para o problema, verificou-se o tempo de treinamento para cada uma dessas combinações.

Como base de dados para o treinamento, utilizaram-se imagens obtidas de pessoas voluntárias a este projeto, em um ambiente com iluminação controlada. Foram consideradas duas pessoas registradas na base de dados, treinando um SVR para cada uma, onde as amostras positivas eram as amostras da própria pessoa e as negativas, da outra pessoa dentro da base de dados (abordagem Um Contra Todos convencional). Como base de dados de teste, usou-se 30 imagens positivas pertencentes a pessoa modelada como classe positiva, e 30 imagens negativas, pertencentes a pessoa modelada como classe negativa no treinamento e a pessoas desconhecidas (cinco indivíduos).

Como medida de desempenho, observou-se principalmente a acurácia. O SVR deve retornar valores próximos a um para amostras de testes da pessoa modelada como classe positiva e valores próximos a zero para amostras de testes da pessoa modelada como classe negativa. Para pessoas desconhecidas, os valores devem estar entre 0 e 1, mas não acima do valor limiar da classe positiva. Como verdadeiros positivos, considerou-se as amostras da pessoa registrada na base de dados corretamente identificadas. Como falsos positivos, considerou-se as amostras tanto da pessoa modelada como classe negativa quanto das pessoas desconhecidas que foram identificadas como positivas. Como verdadeiros negativos, considerou-se as amostras corretamente identificadas da pessoa da classe negativa e das pessoas desconhecidas. E como falsos negativos, considerou-se as amostras da classe

positiva erroneamente identificadas.

Como *features*, utilizou-se o LBP sem a normalização da iluminação.

Os resultados se encontram no gráfico a seguir:

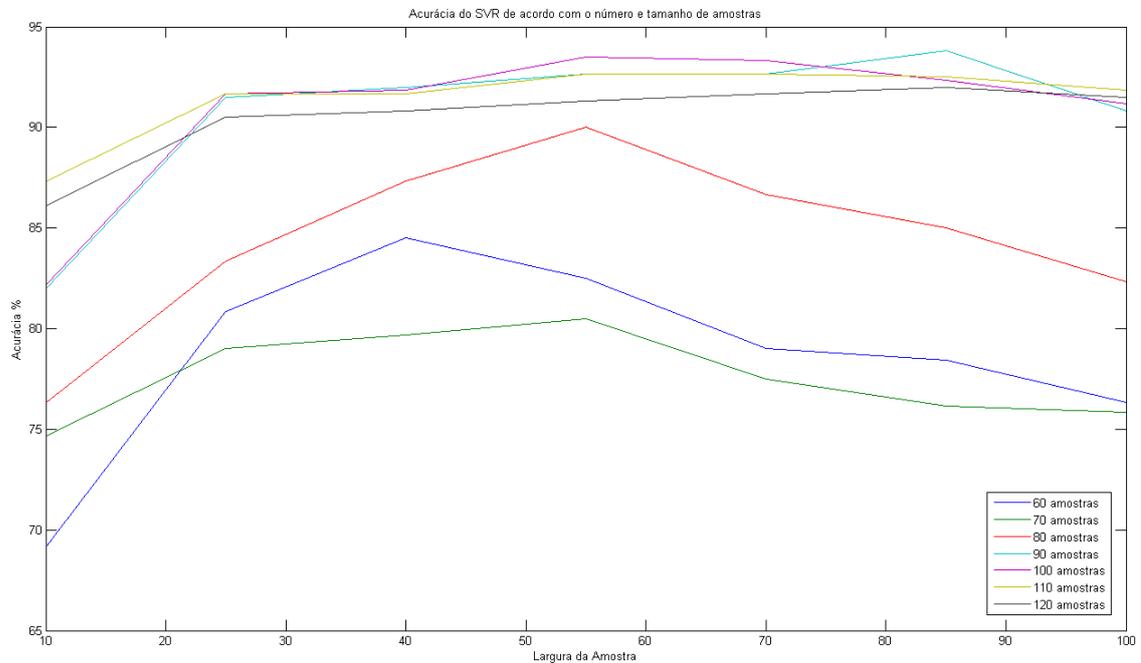


Figura 42 – Acurácia do SVR de acordo com o número e tamanho das amostras.

Como se pode observar no gráfico acima, percebe-se que quanto menos amostras utilizadas, pior é a acurácia do sistema. Entretanto, percebe-se que utilizar 120 amostras ocasiona uma queda no desempenho. Utilizar muitas amostras para o treinamento pode saturar o SVR, ocasionando um possível *overfitting* sobre as imagens de treinamento (ou seja, pouca generalização). Percebe-se que o SVR alcança sua maior acurácia quando a largura da amostra é de 85 pixels e o número de amostras de treinamento é 90.

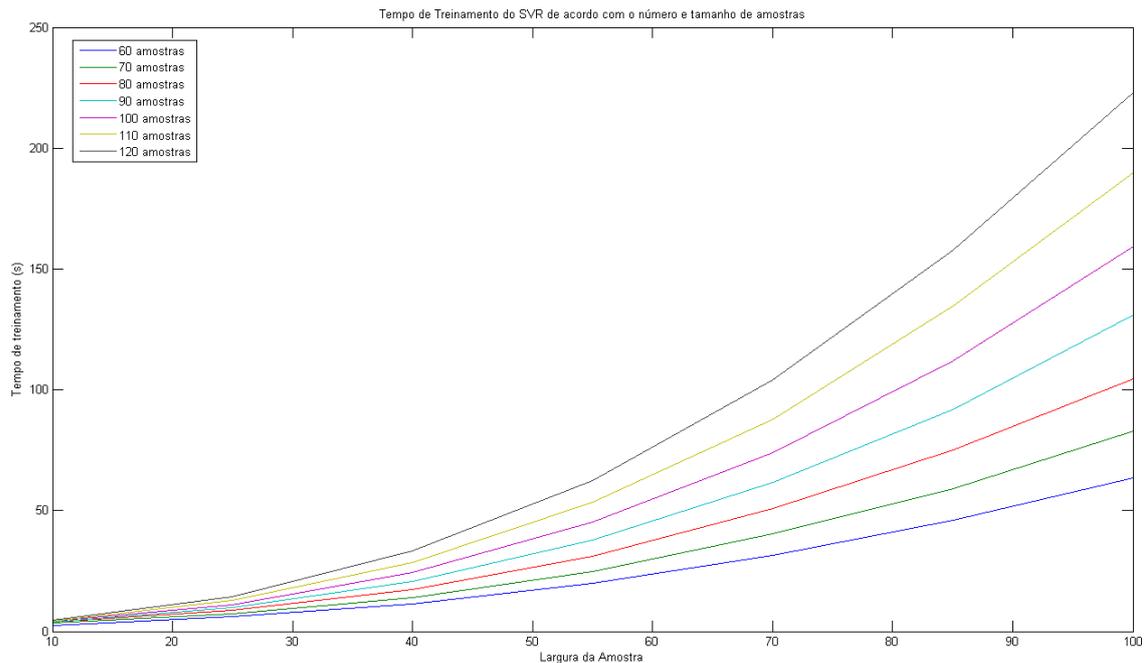


Figura 43 – Tempo de Treinamento do SVR de acordo com o número e tamanho das amostras.

Entretanto, observando os tempos de treinamento do SVR no gráfico acima, percebe-se que utilizar amostras nestas configurações torna o treinamento extremamente lento (cerca de 90 segundos). Considerando que no reconhecedor, estaremos treinando um SVR para cada classe a ser registrada, esse tempo de treinamento se torna inviável, quando o número de classes for alto.

Voltando para o gráfico de acurácia, percebe-se que utilizar 70, 80 ou 90 amostras, com tamanhos variando de 25 a 100 não tem muito efeito sobre a acurácia do SVR. Desta forma, é preciso verificar também os outros parâmetros de desempenho. Um ponto que é muito importante é a especificidade do SVR, pois considerada a aplicação é desejado que nenhuma pessoa que não esteja cadastrada na base de dados seja reconhecida como parte dela.

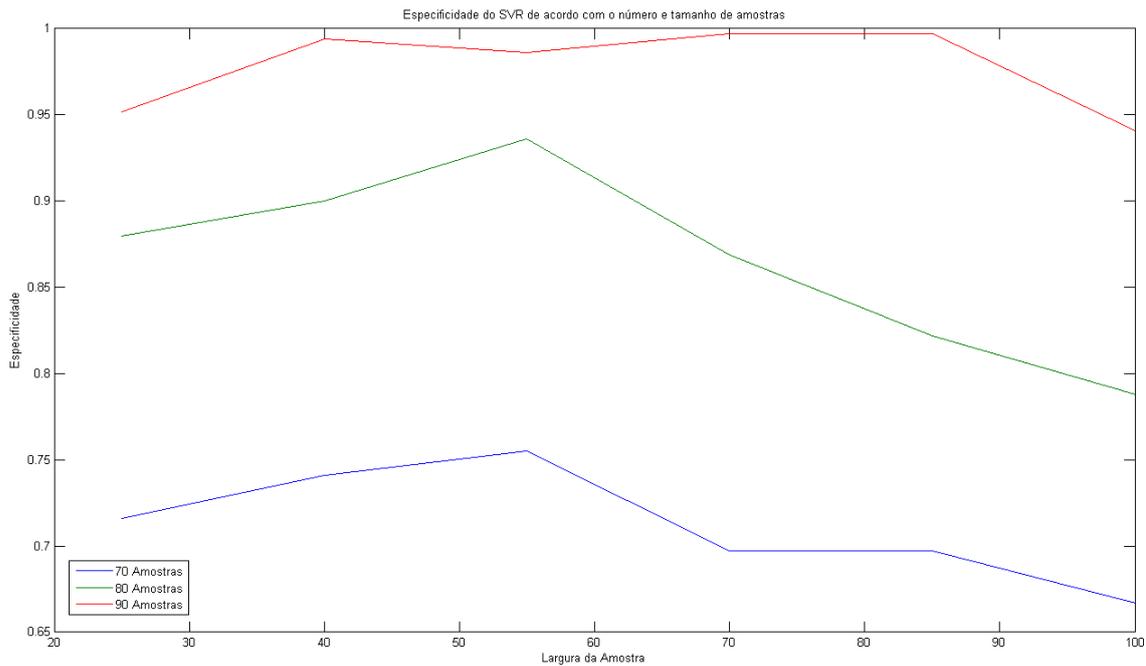


Figura 44 – Especificidade do SVR de acordo com o número e tamanho de amostras

Quanto a maior a especificidade, maior a taxa de reconhecimento de amostras negativas. De fato, a utilização de 90 amostras apresenta uma melhor especificidade, entretanto, percebe-se que é possível alcançar valores de especificidade similares com amostras de tamanho menor ao de 85 pixels. Pelo gráfico dos tempos de treinamento, vê-se que quanto menor a amostra, menor também o tempo. Assim, vê-se que a melhor combinação de número de amostras e tamanho é de 90 amostras de 40 pixels de largura.

3.3.2 Features

Como visto no teste anterior, a utilização dos Padrões Binários Locais, juntamente com a escolha mais apropriada de tamanho e número de amostras, leva a um bom reconhecimento por parte do SVR. Entretanto, para validar a maior eficácia desses *features*, realizou-se um teste treinando o SVR com amostras utilizando apenas o nível de cinza da imagem, com e sem a normalização da iluminação. Testou-se também o desempenho do SVR utilizando os LBP juntamente com a normalização.

Usando a mesma base de dados do teste anterior, treinou-se os SVRs, utilizando como características os níveis de cinza da imagem sem nenhum tipo de tratamento.

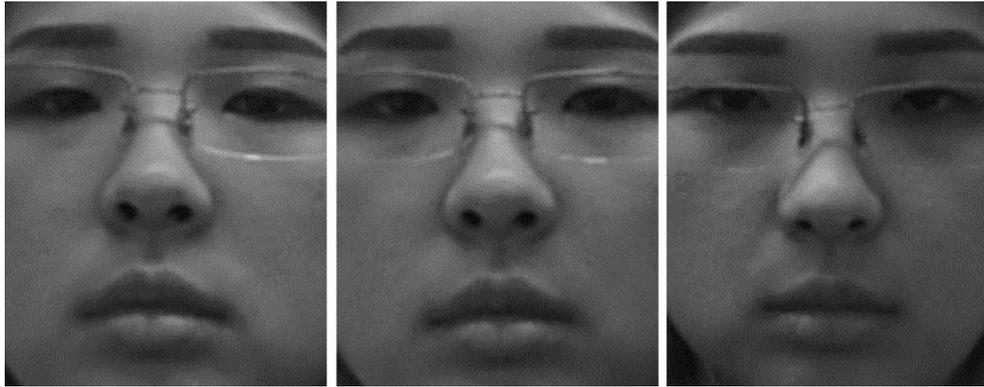


Figura 45 – Imagens em nível de cinza usadas no treinamento do SVR.

Fonte: Autoria própria.

Os seguintes resultados foram obtidos:

Reconhecedor de Rosto - Nível de Cinza	
Parâmetro	Porcentagem
Precisão	90,47
Sensibilidade	47,50
Especificidade	95,00
Acurácia	71,25

Tabela 7 – Reconhecedor de Rosto - Utilizando Nível de Cinza.

Observando os valores obtidos, percebe-se que apesar da alta precisão, o que indica que do total das amostras consideradas como positivas, mais de 90% foi corretamente reconhecida, a sensibilidade é muito baixa, o que mostra que este reconhecedor considerou muitas amostras de pessoas conhecidas como desconhecidas. Ainda que a especificidade tenha sido alta, ter uma baixa porcentagem de sensibilidade é inconveniente, pois na aplicação, somente em 47,50% das vezes as pessoas registradas iriam conseguir entrar na sala.

Importante observar que mesmo estando em um ambiente com iluminação controlada e todas as três imagens serem frontais, existe uma pequena variação em como a pessoa se posiciona diante a câmera, o que acarreta uma mudança, ainda que sutil, na iluminação do rosto. Devido a isto, a utilização de apenas os níveis de cinza da imagem não produz altas taxas de desempenho.

Testou-se novamente, com a mesma base de dados, mas considerando como característica o nível de cinza da imagem com a iluminação normalizada.



Figura 46 – Imagens com iluminação normalizada usadas no treinamento do SVR.

Fonte: Autoria própria.

Obtiveram-se os seguintes resultados:

Reconhecedor de Rosto - Nível de Cinza da Imagem normalizada	
Parâmetro	Porcentagem
Precisão	74,46
Sensibilidade	87,50
Especificidade	80,00
Acurácia	78,75

Tabela 8 – Reconhecedor de Rosto - Utilizando Nível de Cinza da Imagem normalizada.

Aplicar a normalização da iluminação nas imagens acarretou no aumento da acurácia, ou seja este reconhecedor identificou corretamente, em geral, mais pessoas do que o reconhecedor treinado com os níveis de cinza da imagem sem a normalização. A sensibilidade também aumentou consideravelmente, indicando que as pessoas registradas na base de dados seriam autorizadas a entrar na sala mais vezes. Entretanto, houve uma queda na especificidade, o que mostra que mais pessoas desconhecidas também seriam autorizadas a entrar na sala.

Verificou-se, então, o reconhecedor treinado utilizando os padrões binários locais como características, sem a normalização da iluminação.

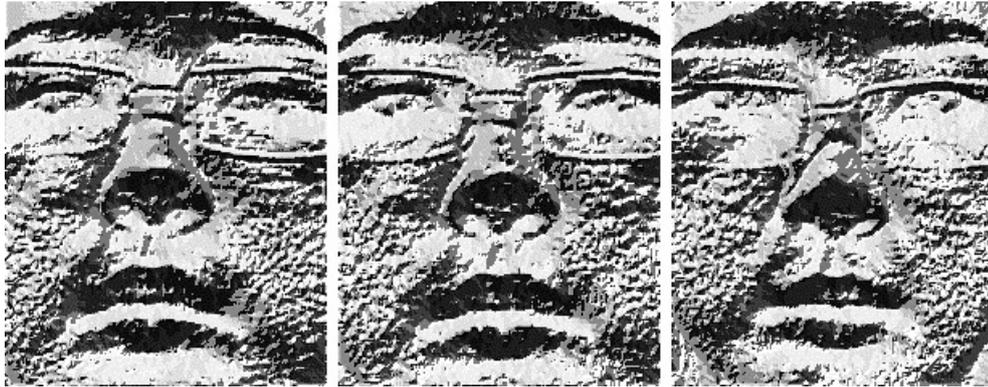


Figura 47 – Imagens com LPB usadas no treinamento do SVR.

Fonte: Autoria própria.

Os seguintes resultados foram obtidos:

Reconhecedor de Rosto - Padrões Binários Locais	
Parâmetro	Porcentagem
Precisão	92,31
Sensibilidade	90,00
Especificidade	92,50
Acurácia	91,25

Tabela 9 – Reconhecedor de Rosto - Utilizando Padrões Binários Locais.

Claramente, a utilização do LBP como característica para o treinamento do SVR melhorou a performance do reconhecedor em todos os aspectos. Comparado ao reconhecedor treinado apenas com níveis de cinza, apesar de ter a especificidade mais baixa, todos os outros parâmetros se mostram melhores. O mesmo percebe-se ao comparar os resultados com a utilização dos níveis de cinza da imagem normalizada. Isso mostra que os padrões locais binários, quando as amostras são obtidas em um ambiente de iluminação controlada, se mostram mais eficientes na caracterização dos rostos, levando a uma maior taxa de reconhecimento.

Vendo que a aplicação da normalização da iluminação nas imagens aumentou o desempenho geral do reconhecedor em relação a utilização dos níveis de cinza sem a mesma, pensou-se na possibilidade de um aumento no desempenho do reconhecedor com a utilização do LBP mais a normalização. Desta forma, realizou-se mais um teste com estas características e os seguintes resultados foram obtidos:

Entretanto, ao se observar os resultados, percebe-se que, apesar de apresentar uma sensibilidade igual ao SVR treinado com o LBP sem a normalização, todas as outras medidas mostraram uma grande queda, principalmente, na especificidade. Comparando com os outros testes, vê-se que a utilização do LBP com a normalização só é vantajoso

Reconhecedor de Rosto - LBP e Normalização da Iluminação	
Parâmetro	Porcentagem
Precisão	73,47
Sensibilidade	90,00
Especificidade	67,50
Acurácia	78,75

Tabela 10 – Reconhecedor de Rosto - Utilizando Padrões Binários Locais e a Normalização da Iluminação.

sobre a utilização dos níveis de cinza puros. Isso se deve ao fato de que o LBP é um potente extrator de características baseado na textura das superfícies. Ao aplicar a normalização da luz, grande parte da imagem do rosto se torna praticamente uniforme, resultando em uma grande geração de ruídos, como se pode ver nas imagens abaixo:

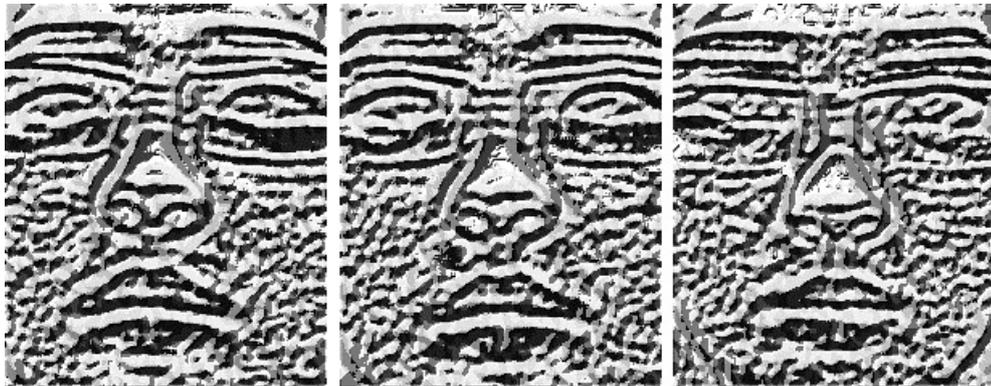


Figura 48 – Imagens com iluminação normalizada e LBP usadas no treinamento do SVR.

Fonte: Autoria própria.

Desta forma, percebe-se que, apesar da grande eficiência do algoritmo de normalização da iluminação, ele não é compatível com a utilização dos Padrões Binários Locais. Entretanto, observando as etapas da normalização, observa-se que a maior parte da uniformização da textura da imagem ocorre depois que se aplica o segundo passo do algoritmo, o filtro diferença de gaussiana. Até então, tem-se apenas a aplicação do filtro Gamma que é eficiente no clareamento das áreas mais escuras e escurecimento das áreas mais claras da imagem, efeitos que são suavizados pelo LBP, não são totalmente eliminados. Sendo assim, testou-se o treinamento do SVR com a aplicação do filtro gamma juntamente com a extração do LBP.



Figura 49 – Imagens com filtro Gamma e LBP usadas no treinamento do SVR.

Fonte: Autoria própria.

A diferença entre estas imagens e as com apenas o LBP é sutil, mas percebe-se uma uniformização na textura do nariz, testa e bochechas. Os resultados obtidos são os seguintes:

Reconhecedor de Rosto Filtro Gamma e LBP	
Parâmetro	Porcentagem
Precisão	97,30
Sensibilidade	90,00
Especificidade	96,50
Acurácia	93,75

Tabela 11 – Reconhecedor de Rosto - Utilizando filtro Gamma e LBP.

E apesar da pequena diferença visual nas imagens utilizadas para o treinamento, percebe-se um aumento em todas as medidas, quando comparado com o reconhecedor treinado apenas com o LBP. Isso mostra que, mesmo sendo invariante a variações monotônicas de intensidade, o que ajuda no aumento da performance do SVR, o LBP por si só não consegue superar os problemas de sombras e ruídos em uma imagem de rosto e, desta forma, o filtro gamma consegue contribuir na melhora do desempenho do reconhecedor.

A seguir, um gráfico mostrando a acurácia para cada um dos *features* utilizados:

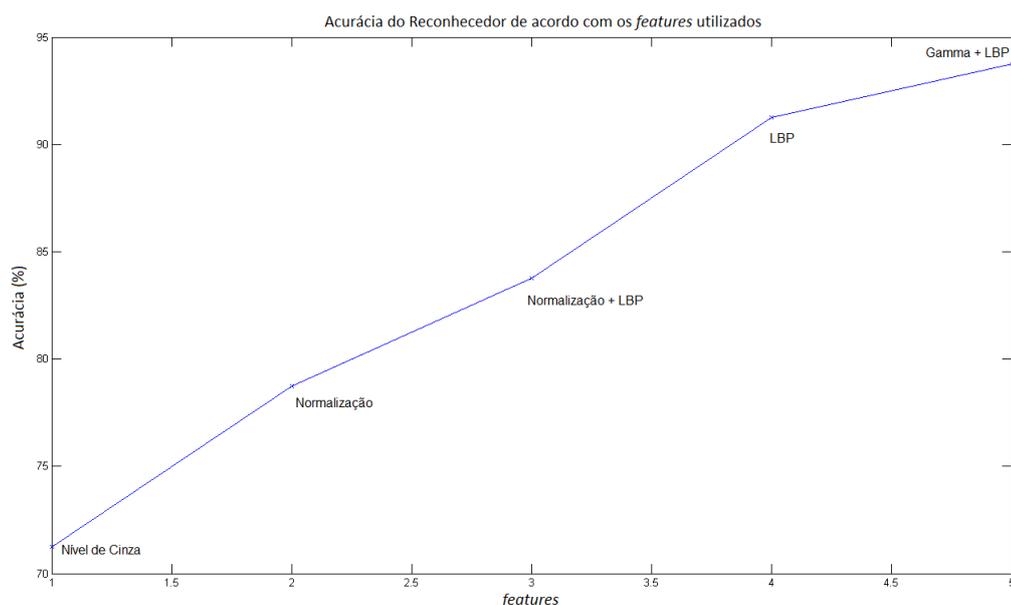


Figura 50 – Gráfico comparativo da acurácia do reconhecedor treinado com diferentes características.

Fonte: Autoria própria.

3.3.3 Número de pessoas registradas

Até o momento, tratamos apenas de um reconhecedor de face com duas pessoas registradas na base de dados. É preciso averiguar como o número de pessoas dentro da base de dados pode influenciar no desempenho do reconhecedor. Para tanto, treinou-se um reconhecedor para duas, três, quatro e cinco pessoas registradas na base de dados, utilizando a abordagem Um Contra Todos, verificando as medidas de desempenho e o tempo de treinamento total do reconhecedor.

Obtiveram-se os seguintes resultados:

Desempenho do Reconhecedor - Um Contra Todos convencional				
	Precisão	Sensibilidade	Especificidade	Acurácia
Duas pessoas	97,30	90,00	96,50	93,75
Três pessoas	95,83	76,67	96,67	86,67
Quatro pessoas	95,31	76,25	96,85	86,25
Cinco pessoas	95,77	68,00	97,00	82,50

Tabela 12 – Desempenho do reconhecedor de acordo com o número de pessoas registradas - Um Contra Todos convencional.

Conforme o número de pessoas registradas aumenta, a taxa de precisão se mantém em torno dos 95% e a de especificidade cresce gradativamente. Esse aumento é esperado, pois conforme o número de pessoas na base de treinamento aumenta, tem-se mais amostras

negativas para representar a classe negativa. Com isso, consegue-se determinar um limiar de classe mais preciso. Entretanto, taxa de sensibilidade cai consideravelmente, afetando diretamente a acurácia do reconhecedor. Isso mostra que o reconhecedor começa a encontrar dificuldade em prever a identidade de uma pessoa registrada. Isso é explicado ao crescente desbalanceamento da base de dados de treinamento.

Treinando os reconhecedores utilizando a abordagem Um Contra Todos balanceada, obtiveram-se os seguintes resultados:

Desempenho do Reconhecedor - Um Contra Todos Balanceado				
	Precisão	Sensibilidade	Especificidade	Acurácia
Duas pessoas	94,44	85,00	95,00	90,00
Três pessoas	96,15	83,33	96,67	90,00
Quatro pessoas	98,48	81,25	98,75	90,00
Cinco pessoas	98,68	75,00	99,00	87,00

Tabela 13 – Desempenho do reconhecedor de acordo com o número de pessoas registradas - Um Contra Todos balanceado.

Ao analisar estes resultados, encontra-se o mesmo comportamento de que quando usada a abordagem convencional, entretanto, a sensibilidade cai em uma taxa menor e a especificidade aumenta em uma taxa maior, conseguindo manter a acurácia do reconhecedor, mostrando que a abordagem proposta por este trabalho ajuda a melhorar o desempenho do reconhecedor.

Além disso, calculou-se o tempo total de treinamento para as duas formas da abordagem:

Tempo de Treinamento - Convencional x Proposta		
	Convencional	Proposta
Duas pessoas	49,48	50,43
Três pessoas	126,28	70,99
Quatro pessoas	260,81	93,03
Cinco pessoas	460,82	113,52

Tabela 14 – Tempo de Treinamento para as duas formas da abordagem Um Contra Todos em (s).

Para duas pessoas, a diferença de tempo é pequena pois nesta condição, o tamanho da base de treinamento é o mesmo (já que só há duas pessoas na base de dados, usar todas as amostras já gera uma base balanceada). Entretanto, para os outros números de pessoas, percebe-se uma grande diferença nos tempos de treinamento, com o tempo de treinamento da abordagem convencional alcançando até quatro vezes maior do que abordagem proposta.

Sendo assim, considerando as medidas de desempenho do reconhecedor e o seu tempo de treinamento, a abordagem proposta pelo trabalho se mostrou mais eficiente.

3.3.4 Predição a partir de vários quadros

Para diminuir os erros na predição, propôs-se a verificação da identidade da pessoa em diversos quadros consecutivos, registrando as *label* retornadas pelo reconhecedor e identificando a pessoa com a *label* que for retornada com mais de 85% dos votos. Para verificar como isso melhora no desempenho do reconhecedor, executou-se um teste, processando vários quadros de vídeos, verificando em média quantos quadros foram necessários para a decisão da *label*:

Número médio de quadros	
Duas pessoas	6,41
Três pessoas	7,69
Quatro pessoas	7,41
Cinco pessoas	6,10

Tabela 15 – Número médio de quadros analisados para tomar uma decisão.

E analisando os valores retornados depois a averiguação dos votos, conseguiu-se as seguintes medidas, que representam o desempenho do reconhecedor em sua totalidade implementado no sistema:

Medidas de desempenho para o reconhecedor				
	Precisão	Sensibilidade	Especificidade	Acurácia
Duas pessoas	98,90	96,55	99,92	98,03
Três pessoas	99,00	93,85	99,89	94,73
Quatro pessoas	98,23	87,87	98,92	90,09
Cinco pessoas	99,30	91,42	98,87	94,00

Tabela 16 – Medidas de desempenho para o reconhecedor.

Observando-se esses resultados, percebe-se que a utilização de mais de um quadro na decisão da predição da classe de uma pessoa aumenta significativamente o desempenho do reconhecedor. É importante ressaltar a alta porcentagem de especificidade, o que mostra que o reconhecedor obtido consegue identificar se uma amostra é desconhecida em cerca de 99% das vezes que uma amostra de uma pessoa desconhecida é testada.

3.4 Sistema Implementado

O sistema de reconhecimento facial proposto por este trabalho foi implementado utilizando a linguagem C++, com a interface criada de forma independente usando a linguagem Qt. O *software* foi nomeado Megumi, junção de duas palavras japonesas "Me" que significa olho e "gumi" de "bangumi" que significa programa (esta é uma tradução extremamente literal e, na verdade, não irá fazer muito sentido a qualquer pessoa que realmente domine a língua japonesa). Este programa estabelece uma comunicação serial via Zigbee com um pequeno circuito eletrônico para o acionamento de uma trava elétrica. Caso a pessoa a ser reconhecida seja identificada como alguém de dentro da base de dados, o programa envia um sinal liberando a trava.

3.4.1 Software

A tela principal do programa ficou da seguinte maneira:

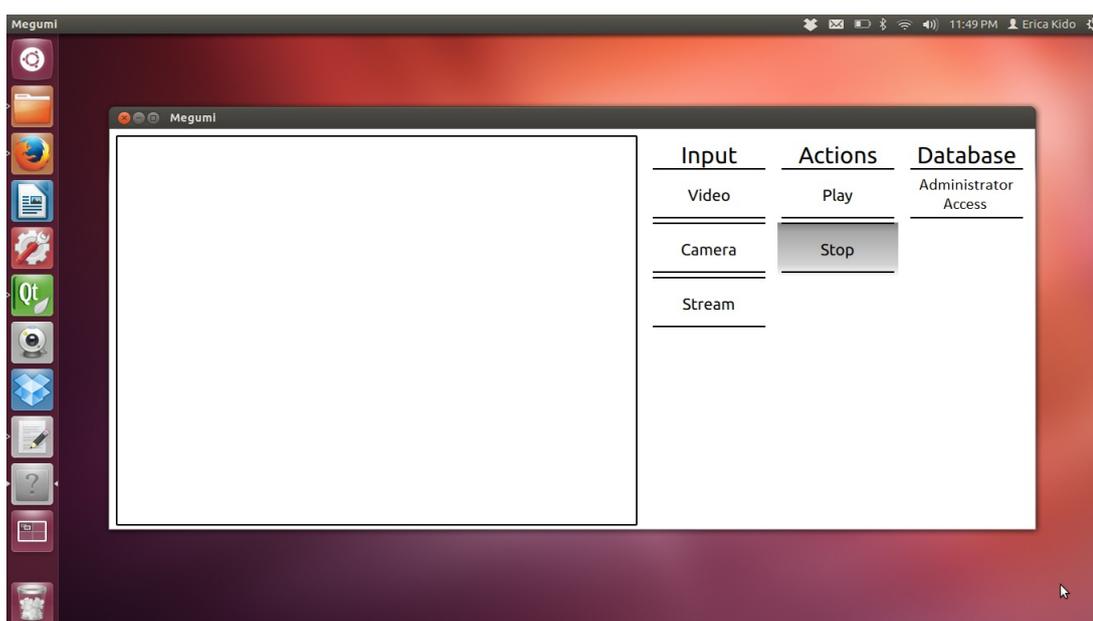


Figura 51 – Megumi - Tela Principal.

Fonte: Autoria própria.

Ela apresenta uma área branca onde será mostrada a sequência de quadros que pode ser obtido de um arquivo de vídeo, de uma câmera conectada ao computador ou de um *stream* via *wi-fi*.

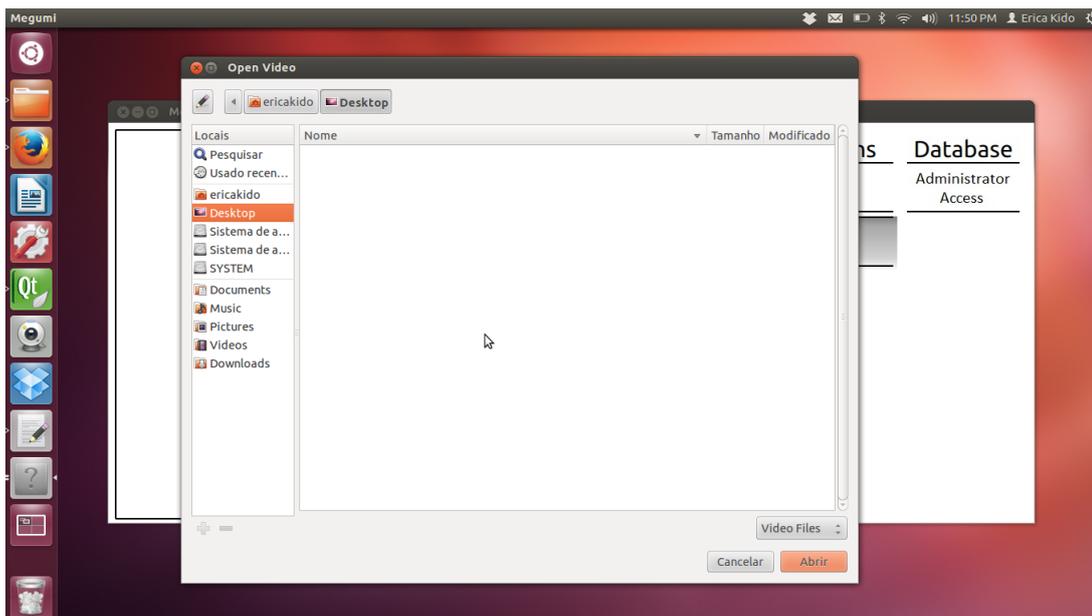


Figura 52 – Megumi - Carregar vídeo de um arquivo.

Fonte: Autoria própria.

Ao escolher um dos métodos de entrada para o vídeo, apertando a opção *play* inicia a exibição do vídeo e o reconhecimento facial.

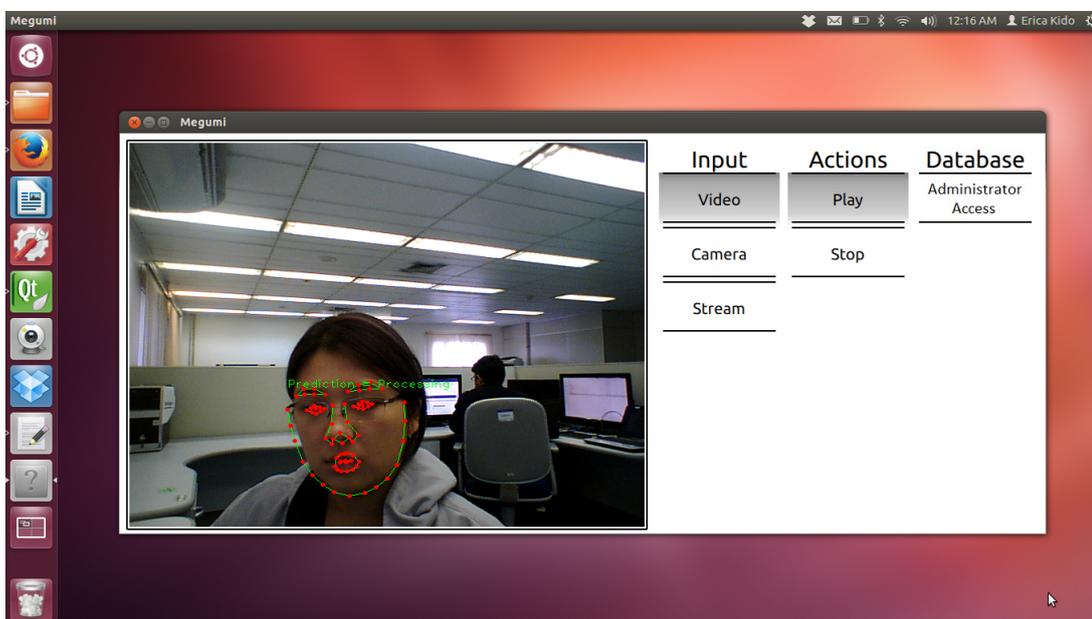


Figura 53 – Megumi - Reconhecimento.

Fonte: Autoria própria.

O menu *Database* só pode ser acessado pelo administrador. Ao clicar em *Administrator Access*, uma tela de *log in* aparece:

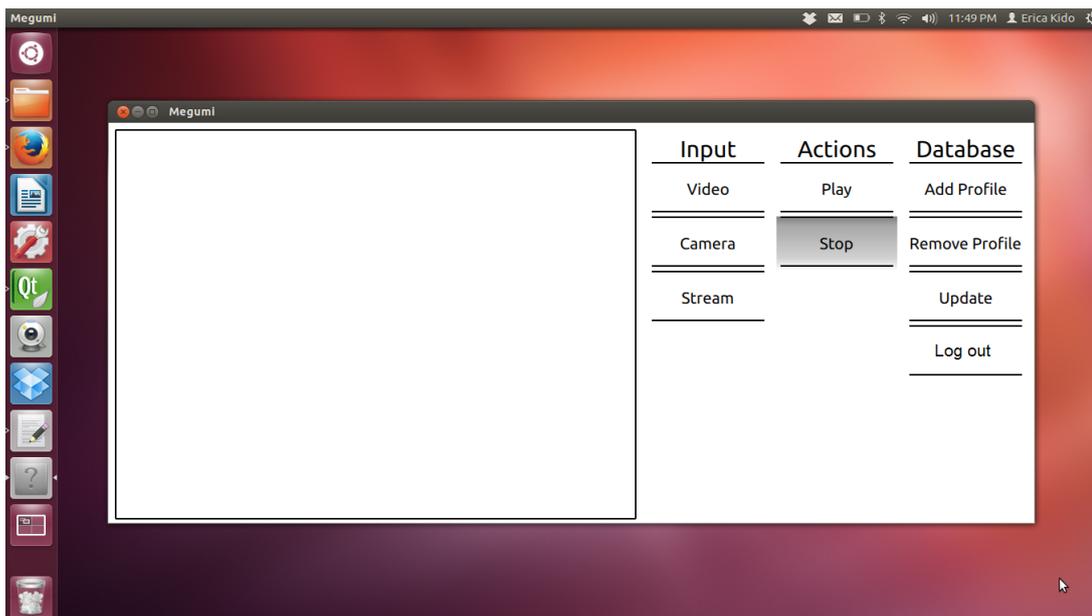


Figura 55 – Megumi - Tela com acesso de administrador.

Fonte: Autoria própria.

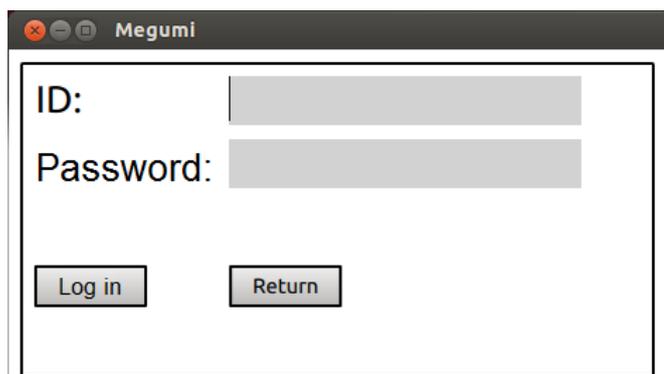


Figura 54 – Megumi - Tela de *log in*.

Fonte: Autoria própria.

O administrador é cadastrado na primeira vez que o programa é executado. Ao acessar inserir os dados do administrador, o menu da *Database* é liberado:

Nestas opções, é possível adicionar um perfil novo, ou seja, adicionar uma nova pessoa a base de dados:

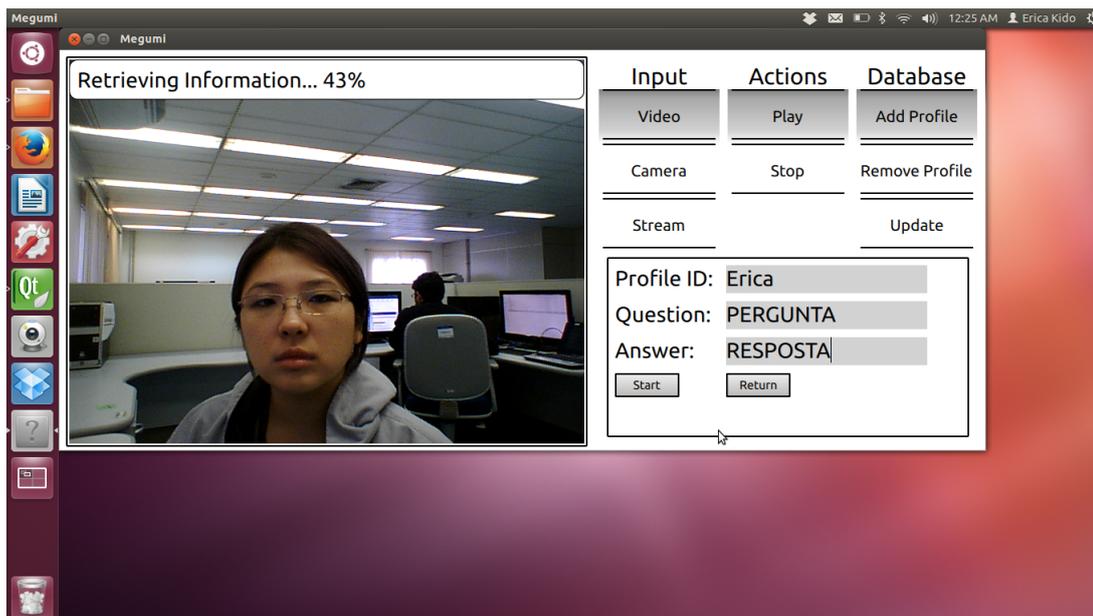


Figura 57 – Megumi - Obtendo amostras do novo perfil.

Fonte: Autoria própria.

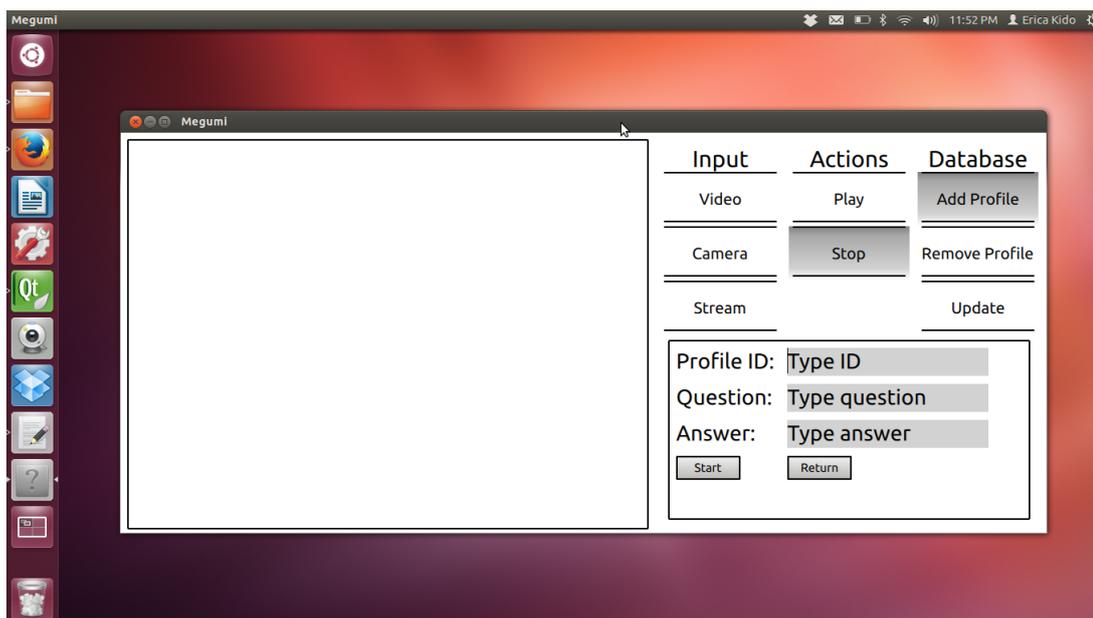


Figura 56 – Megumi - Adicionar novo perfil.

Fonte: Autoria própria.

Ao adicionar um perfil novo, é necessário inserir um nome de identificação para o perfil, a *ID*, uma pergunta e resposta de segurança que é utilizada apenas em momentos de extrema confusão do detector (por exemplo, numa sequência de quadros, em 50% dos casos retornou um perfil, nos outros 50%, outro perfil). Ao clicar em *Start*, inicia o processo de aquisição de amostras do novo perfil:

Após a adição do(s) novo(s) perfil(is), é necessário clicar na opção *Update* para que

seja realizado o treinamento do reconhecedor.

É possível também remover perfis da base de treinamento, ao clicar na opção *Remove Profile*.

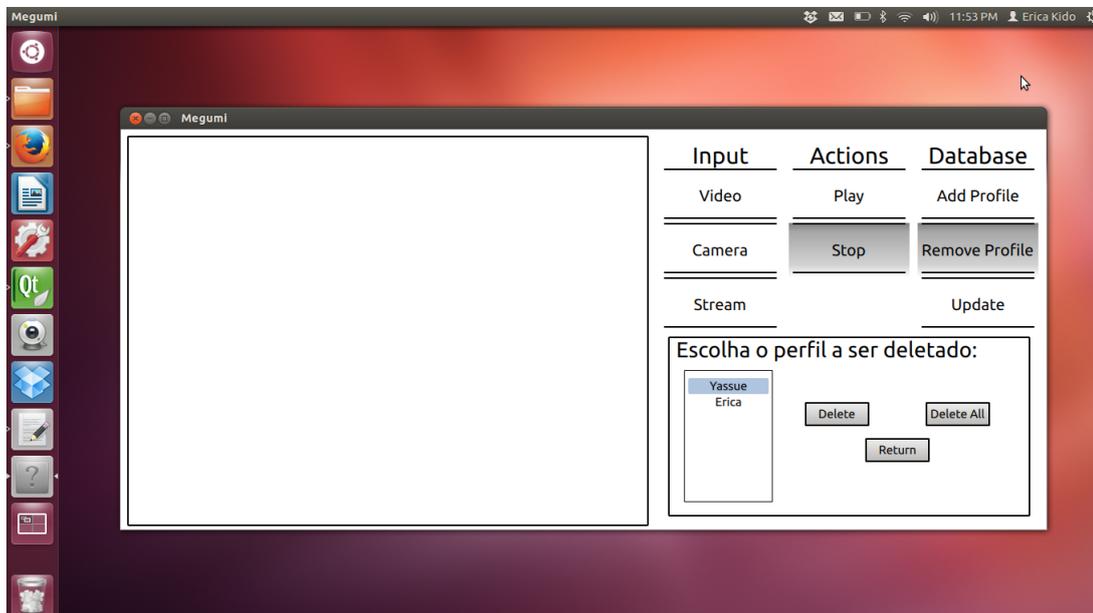


Figura 58 – Megumi - Deletando um perfil.

Fonte: Autoria própria.

Uma lista dos nomes dentro da base de dados é mostrada, podendo-se deletar apenas um dos perfis selecionando o nome do mesmo e clicando no botão *Delete*, ou deletar todos os perfis, clicando em *Delete All*. O botão *Return* oculta a janela de remoção de perfis. Depois que um perfil é deletado, é necessário clicar na opção *Update* para que os reconhecedores sejam retreinados.

3.4.2 Hardware

A parte física implementada ficou da seguinte maneira:

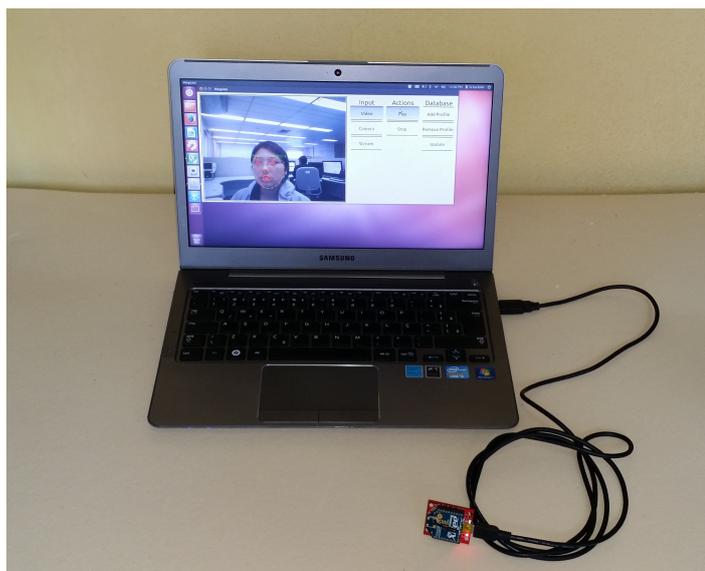


Figura 59 – Computador ligado ao *ZigBee* Coordenador.

Fonte: Autoria própria.

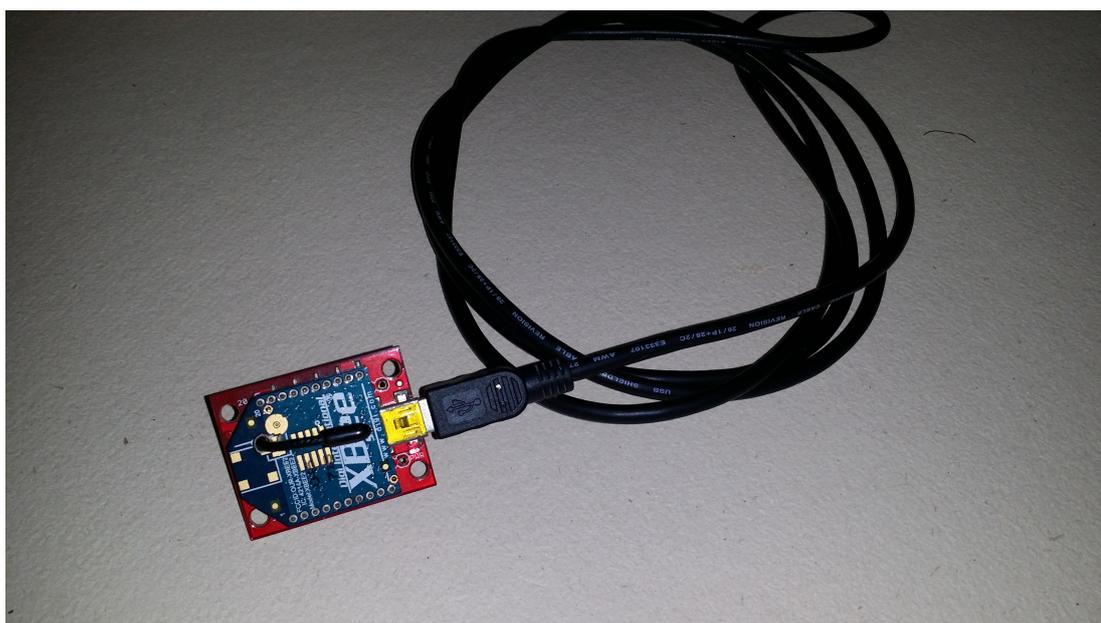


Figura 60 – *ZigBee* Coordenador.

Fonte: Autoria própria.

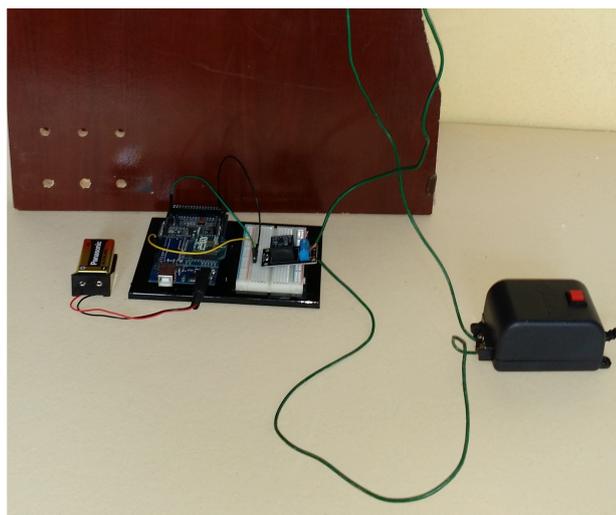


Figura 61 – Controle da fechadura.

Fonte: Autoria própria.

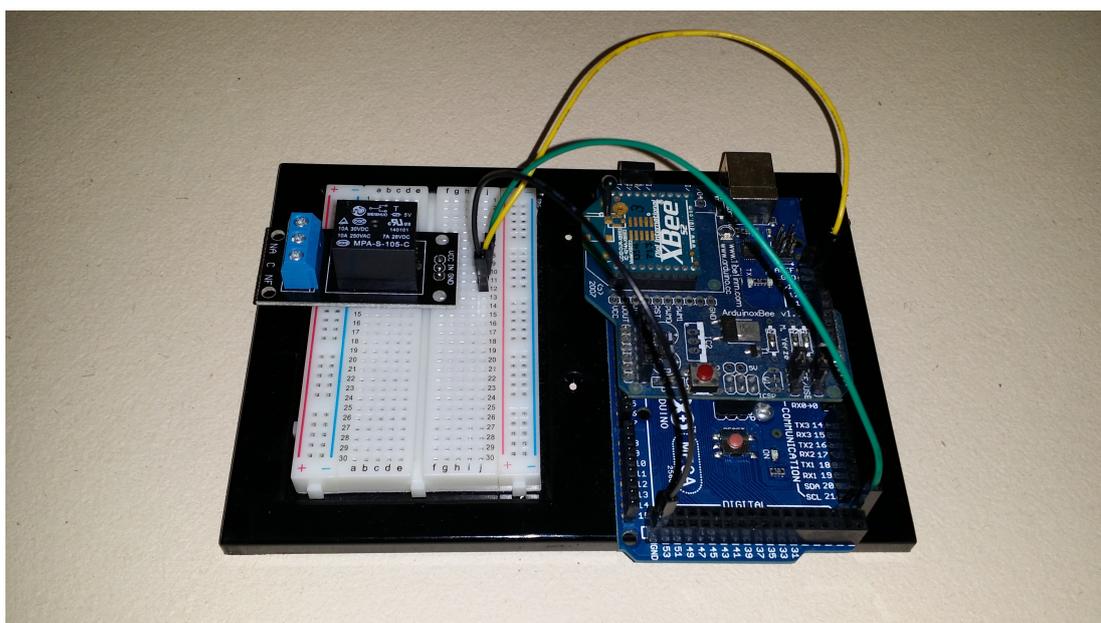


Figura 62 – Microcontrolador ligado ao ZED e ao circuito de acionamento da fechadura.

Fonte: Autoria própria.

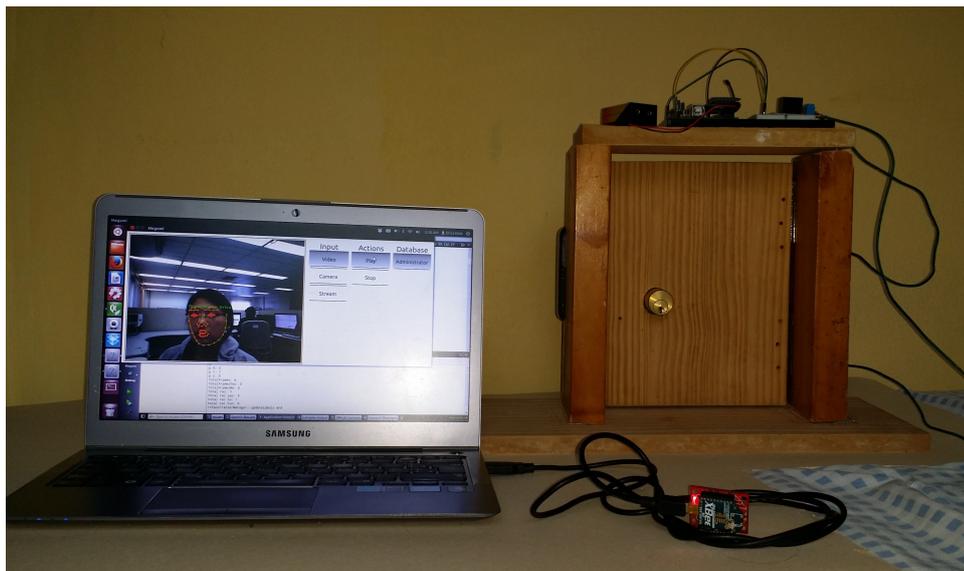


Figura 63 – Parte física completa.

Fonte: Autoria própria.

Para esta implementação foi utilizado o *Arduino* Mega, pois já se possuía este microcontrolador, mas o Uno já seria suficiente. A comunicação via *ZigBee* se mostrou suficiente para a aplicação, uma vez que a distância entre o computador com o programa e o microcontrolador está dentro do alcance dos *XBee*s (menos que 100 metros) e a quantidade de dados a serem transmitidos é pequena.

4 Considerações Finais

NESTE trabalho, foi proposto um novo método para reconhecimento facial em *open-set*. Diante da análise das técnicas já existentes, verificou-se a possibilidade do desenvolvimento de um reconhecer facial utilizando a Regressão por Vetores de Suporte. Dentro deste objetivo geral, propôs-se também um método de detecção facial baseado na segmentação de pixels com cor de pele, com o intuito de reduzir o tempo de execução. Além disso, propôs-se também a criação de um *software* em C++ e sua conexão com um circuito que controla a abertura e fechamento de uma trava elétrica.

Em vista dos resultados obtidos dos testes, pode-se ver a viabilidade dos métodos propostos tanto para detecção facial quanto para reconhecimento.

Quanto ao segmentador de pele, observou-se que apesar de existir uma confusão quando pixels que não correspondem a pele apresentam uma cor semelhante, o número de falsos negativos é pequeno quando comparado ao número de verdadeiros positivos. A utilização do espaço de cores YCrCb se mostrou eficiente para a obtenção dos histogramas de cores, gerando boas detecções de pixel de cor de pele em diversos ambientes encontrados nas imagens da base de dados de teste utilizada.

Quanto ao detector, viu-se que o maior problema está detecção dos elementos do rosto. É preciso então ou mudar as restrições geométricas verificadas para que não ocorra a identificação de um rosto com dois olhos no mesmo lado do rosto, ou procurar um outro método para a identificação dos elementos. Considerando que na aplicação, em geral, a pessoa estará olhando diretamente para a câmera, o erro de encontrar dois olhos no mesmo lado do rosto será menos provável de ocorrer. Sendo o sistema aplicado em um local fixo, é possível controlar o ambiente para que não haja objetos com tom similar ao da pele no plano de fundo da pessoa a ser detectada e, assim, consegue-se também uma melhora na taxa de segmentação da pele.

Mesmo em face a estes problemas, conseguiu-se um detector facial que é, em média, cinco vezes mais rápido do que o reconhecedor que utiliza apenas os classificadores Haar em Cascata.

Quanto ao reconhecedor, viu-se que é possível fazer o reconhecimento facial usando o SVR, com uma grande taxa de identificação tanto de pessoas registradas como de pessoas não registradas na base de dados. Entretanto, observou-se a queda no desempenho do reconhecedor conforme o número de pessoas registradas cresce, o que o torna inviável em aplicações onde muitas pessoas necessitam ser identificadas. Viu-se também que as características a serem utilizadas para o treinamento, assim como o modo como a base de dados é montada influencia diretamente na performance do reconhecedor e que a utilização de vários quadros para a identificação da classe da pessoa aumenta significativamente as medidas de desempenho.

O programa juntamente com a parte física, se mostrou funcional e cumpriu o objetivo de simular o controle de acesso de uma sala.

Viu-se, portanto, que o sistema criado com os métodos propostos neste trabalho atingiram os objetivos propostos, seguindo às suposições apresentadas pelo mesmo.

4.1 Trabalhos Futuros

Sendo este trabalho composto de diversas partes, vê-se uma grande possibilidade de trabalhos futuros. Apesar dos resultados promissores, o sistema ainda não apresenta um desempenho aceitável para aplicações reais. Dentre as melhorias a serem feitas, pode-se citar:

- Reduzir o número de falsos positivos do segmentador de pele: Ainda que seja melhor que o segmentador encontre áreas de pele a mais que de menos, para o desempenho é importante que este processo seja feito da forma mais eficiente possível e para tanto a redução dos falsos positivos e negativos é primordial. Uma forma de reduzir estes números é incrementar ou modificar o banco de imagens utilizados para conseguir os histogramas de cores. Além disso, também é possível melhorar a generalização das cores aproximando a curva encontrada nos histogramas para uma gaussiana. O único problema nisto está no custo computacional adicional que este processo pode acarretar e que pode levar a um aumento no tempo de segmentação.
- Melhorar a detecção dos olhos, nariz e boca: Sendo esta detecção o maior problema na detecção do rosto, vê-se a possibilidade da utilização ou criação de outro método para a detecção dos mesmos.
- Aumentar a robustez do sistema às variações de iluminação: Mudanças nas condições de luz na hora da captura de amostras de treinamento e teste são as maiores causas de erro por parte dos reconhecedores e classificadores de imagem em geral. Encontrou-se um bom método para a normalização da luz, mas percebeu-se que ele é incompatível com os Padrões Binários Locais, pois estes possuem apenas um valor de *threshold* fixo (o pixel central da vizinhança), causando grande inserção de ruídos em regiões quase uniformes. O LBP possui diversas variantes, entre elas sua extensão à utilização da lógica fuzzy, de modo a estabelecer uma faixa de valores que podem ser considerados como um só, reduzindo assim os ruídos. A utilização desta extensão se mostra uma grande candidata na união da normalização da luz com o LBP.

Enfim, todos os elementos definidos podem ser otimizados por meio de estudos específicos, tendo-se assim um grande número de possibilidades para trabalhos futuros.

Referências

- ARDUINO official webpage. 2014. Disponível em: <<http://www.arduino.cc/>>. Citado na página 40.
- BAGGIO, D. L. *Mastering OpenCV with practical computer vision projects*. [S.l.]: Packt Publishing Ltd, 2012. Citado 2 vezes nas páginas 61 e 62.
- BARONTI, P. et al. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards. *Computer communications*, Elsevier, v. 30, n. 7, p. 1655–1695, 2007. Citado na página 34.
- BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. Eigenfaces vs. fisherfaces. recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 19, n. 7, p. 711–720, 1997. Citado na página 25.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, p. 27:1–27:27, 2011. Software available at <<http://www.csie.ntu.edu.tw/~cjlin/libsvm>>. Citado na página 29.
- CHANG, W. F. H.; ROBLES, U. *Face Detection*. 2000. Disponível em: <<http://www-cs-students.stanford.edu/~robles/ee368/main.html>>. Citado na página 20.
- CRAMMER, K.; SINGER, Y. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, JMLR. org, v. 2, p. 265–292, 2002. Citado na página 56.
- EKENEL, H. K.; SZASZ-TOTH, L.; STIEFELHAGEN, R. Open-set face recognition-based visitor interface system. In: *Computer Vision Systems*. [S.l.]: Springer, 2009. p. 43–52. Citado na página 56.
- FARAG, A.; MOHAMED, R. M. Regression using support vector machines: basic foundations. Citeseer, 2004. Citado na página 12.
- FBS Eletrônica. 2014. Disponível em: <<http://www.fbseletronica.com.br/pd-a2597-modulo-1-rele.html?ct=&p=1&s=1>>. Citado na página 40.
- FEIPENG, L. *YCbCr Color Space. An Intro and its Applications*. 2011. Disponível em: <<http://www.roman10.net/ybcr-color-spacean-intro-and-its-applications/>>. Citado na página 20.
- FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936. Citado na página 25.
- FORD, A.; ROBERTS, A. *Colour Space Conversions*. 2014. Disponível em: <<http://www.poynton.com/PDFs/coloureq.pdf>>. Citado na página 19.
- FREEMAN, W. T.; ADELSON, E. H. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, IEEE Computer Society, v. 13, n. 9, p. 891–906, 1991. Citado na página 22.

- FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. In: SPRINGER. *Computational learning theory*. [S.l.], 1995. p. 23–37. Citado na página 23.
- FRISCHHOLZ, R. W. *The Face Detection: Techniques*. 2014. Disponível em: <<http://www.facedetection.com/facedetection/techniques.htm>>. Citado na página 20.
- FUKUNAGA, K. *Introduction to statistical pattern recognition*. [S.l.]: Academic press, 1990. Citado na página 45.
- GONZALES, R. C.; WOODS, R. E. *Digital Image Processing, 2-nd Edition*. [S.l.]: Prentice Hall, 2002. Citado 3 vezes nas páginas 15, 16 e 17.
- GRGIC MISLAV; DELAC, K. *Face Recognition*. 2007. Disponível em: <<http://www.face-rec.org/general-info/>>. Citado na página 20.
- GUNN, S. R. et al. *Support vector machines for classification and regression*. [S.l.], 1998. Citado na página 30.
- HDL - Uma marca do grupo Legrand. 2014. Disponível em: <<http://www.hdl.com.br/search/content/fechaduras>>. Citado na página 40.
- HEISELE, B.; HO, P.; POGGIO, T. Face recognition with support vector machines: Global versus component-based approach. In: IEEE. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. [S.l.], 2001. v. 2, p. 688–694. Citado na página 23.
- HUANG, G. B. et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. [S.l.], 2007. Citado 3 vezes nas páginas 63, 65 e 69.
- HUNT, R. W. G.; POINTER, M. R.; POINTER, M. *Measuring colour*. [S.l.]: John Wiley & Sons, 2011. Citado na página 19.
- HUTCHINSON, L. *Brazilian docs fool biometric scanners with bag full of fake fingers*. 2013. Disponível em: <<http://arstechnica.com/tech-policy/2013/03/brazilian-docs-fool-biometric-scanners-with-bag-full-of-fake-fingers/>>. Citado na página 12.
- KHAN, R. et al. Color based skin classification. *Pattern Recognition Letters*, Elsevier, v. 33, n. 2, p. 157–163, 2012. Citado na página 43.
- LEE, Y.; LIN, Y.; WAHBA, G. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, Taylor & Francis, v. 99, n. 465, p. 67–81, 2004. Citado na página 56.
- LI, F.; WECHSLER, H. Open set face recognition using transduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 27, n. 11, p. 1686–1697, 2005. Citado na página 11.
- LIENHART, R.; KURANOV, A.; PISAREVSKY, V. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In: *Pattern Recognition*. [S.l.]: Springer, 2003. p. 297–304. Citado na página 21.

- LIU, D.; ZHANG, X.; DING, F.-k. The design of intelligent lighting monitoring system on zigbee based wireless personal area network. In: ATLANTIS PRESS. *2014 International Conference on Mechatronics, Control and Electronic Engineering (MCE-14)*. [S.l.], 2014. Citado na página 34.
- MILBORROW, S.; MORTEL, J.; NICOLLS, F. The MUCT Landmarked Face Database. *Pattern Recognition Association of South Africa*, 2010. <<http://www.milbo.org/muct>>. Citado na página 61.
- MORESCHI, M. *Inspire-se LARANJA*. 2011. Disponível em: <<http://arquiteturadiaria.blogspot.com.br/2011/01/inspire-se-laranja.html>>. Citado na página 64.
- OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In: *Advances in Pattern Recognition - ICAPR 2001*. [S.l.]: Springer, 2001. p. 399–408. Citado na página 25.
- OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, IEEE*, v. 24, n. 7, p. 971–987, 2002. Citado na página 27.
- OPENCV. *Cascade Classification*. 2014. Disponível em: <http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html>. Citado na página 21.
- OPENCV. *Face Recognition with OpenCV*. 2014. Disponível em: <http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html>. Citado na página 25.
- OSUNA, E.; FREUND, R.; GIROSI, F. Training support vector machines: an application to face detection. In: IEEE. *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. [S.l.], 1997. p. 130–136. Citado na página 29.
- PAN, G. et al. Eyeblink-based anti-spoofing in face recognition from a generic webcam. In: IEEE. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. [S.l.], 2007. p. 1–8. Citado na página 20.
- PAPAGEORGIOU, C. P.; OREN, M.; POGGIO, T. A general framework for object detection. In: IEEE. *Computer vision, 1998. sixth international conference on*. [S.l.], 1998. p. 555–562. Citado na página 21.
- PHILLIPS, P. J. et al. Overview of the face recognition grand challenge. In: IEEE. *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*. [S.l.], 2005. v. 1, p. 947–954. Citado na página 11.
- PHUNG, S. L.; BOUZERDOUM, A.; SR, D. C. Skin segmentation using color pixel classification: analysis and comparison. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, IEEE*, v. 27, n. 1, p. 148–154, 2005. Citado na página 45.
- PIETIKAINEN, M. et al. Local binary patterns for still images. In: *Computer Vision Using Local Binary Patterns*. [S.l.]: Springer, 2011. p. 13–47. Citado na página 25.

- QUEIROZ, C. J. d. *Análise de Transformações Geométricas para o Georreferenciamento de Imagens do Satélite CBERS-I*. Dissertação (Mestrado) — UFRGS - CEPSRM, 2003. Citado na página 15.
- REIGNIER, P. *Finding a face by blink detection*. 1995. Disponível em: <<http://www-prima.imag.fr/ECVNet/IRS95/node13.html>>. Citado na página 20.
- RIFKIN, R.; KLAUTAU, A. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, JMLR. org, v. 5, p. 101–141, 2004. Citado na página 56.
- SCHAPIRE, R. E.; SINGER, Y. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, Springer, v. 37, n. 3, p. 297–336, 1999. Citado na página 21.
- SCHEIRER, W. J.; JAIN, L. P.; BOULT, T. E. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, v. 10, p. 1–8, 2014. To appear, preprint at <http://www.wjscheirer.com/papers/wjs_pami2014_probability.pdf>. Citado na página 11.
- SCHEIRER, W. J. et al. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, v. 36, p. 1–16, July 2013. Citado na página 11.
- SHAN, C. Face recognition and retrieval in video. In: *Video Search and Mining*. [S.l.]: Springer, 2010. p. 235–260. Citado na página 21.
- SIROVICH, L.; KIRBY, M. Low-dimensional procedure for the characterization of human faces. *JOSA A*, Optical Society of America, v. 4, n. 3, p. 519–524, 1987. Citado na página 25.
- SMELSER, N. J.; BALTES, P. B. et al. *International encyclopedia of the social and behavioral sciences*. [S.l.]: Elsevier Amsterdam, 2001. Citado na página 19.
- STALLKAMP, J.; EKENEL, H. K.; STIEFELHAGEN, R. Video-based face recognition on real-world data. In: IEEE. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. [S.l.], 2007. p. 1–8. Citado na página 11.
- TAN, K.; CHEN, S. Adaptively weighted sub-pattern pca for face recognition. *Neurocomputing*, Elsevier, v. 64, p. 505–511, 2005. Citado na página 25.
- TAN, X.; TRIGGS, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Image Processing, IEEE Transactions on*, IEEE, v. 19, n. 6, p. 1635–1650, 2010. Citado 2 vezes nas páginas 48 e 52.
- TAX, D. M.; DUIN, R. P. Support vector data description. *Machine learning*, Springer, v. 54, n. 1, p. 45–66, 2004. Citado na página 11.
- VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: springer, 2000. Citado na página 27.
- VAPNIK, V. N.; VAPNIK, V. *Statistical learning theory*. [S.l.]: Wiley New York, 1998. Citado na página 30.

VIOLA, P.; JONES, M. J. Robust real-time face detection. *International journal of computer vision*, Springer, v. 57, n. 2, p. 137–154, 2004. Citado 2 vezes nas páginas 21 e 22.

WESTON, J.; WATKINS, C. *Multi-class support vector machines*. [S.l.], 1998. Citado na página 56.

ZELENOVSKY, R.; MENDONÇA, A. Microcontroladores: programação e projeto com a família 8051. *Rio de Janeiro: MZ*, 2005. Citado na página 32.

ZHANG, C.; ZHANG, Z. *A survey of recent advances in face detection*. [S.l.], 2010. Citado na página 11.

ZOU, K. H.; OMALLEY, A. J.; MAURI, L. Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. *Circulation*, Am Heart Assoc, v. 115, n. 5, p. 654–657, 2007. Citado na página 32.