

UNIVERSIDADE DE SÃO PAULO
Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica e de Computação

**Implementação de um sistema de
identificação facial utilizando Linux
Embarcado**

Victor Nascimento

Orientador: Prof. Dr. Evandro Luis L. Rodrigues

São Carlos
2015

VICTOR NASCIMENTO

IMPLEMENTAÇÃO DE UM SISTEMA DE IDENTIFICAÇÃO FACIAL UTILIZANDO LINUX EMBARCADO

Trabalho de Conclusão de Curso apresentado à Escola de Engenharia de São Carlos como parte dos requisitos para a obtenção do título de Engenharia de Computação.

Área de concentração: Sistemas Embarcados

Orientador: Prof. Dr. Evandro Luís L. Rodrigues

São Carlos

2015

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Nascimento, Victor
N244i Implementação de um sistema de identificação facial
utilizando Linux Embocado / Victor Nascimento;
orientador Evandro Luis Linhari Rodrigues. São Carlos,
2015.

Monografia (Graduação em Engenharia de Computação)
-- Escola de Engenharia de São Carlos da Universidade
de São Paulo, 2015.

1. Sistemas Embocados. 2. Linux Embocado. 3.
Beaglebone. 4. OpenCV. 5. reconhecimento facial. 6.
visão computacional. I. Título.

FOLHA DE APROVAÇÃO

Nome: Victor Nascimento

Título: "Implementação de um sistema de reconhecimento facial utilizando Linux Embocado"

Trabalho de Conclusão de Curso defendido em 26/06/2015.

Comissão Julgadora:

Resultado:

Prof. Associado Evandro Luís Linhari Rodrigues
(Orientador) - SEL/EESC/USP

APROVADO

Prof. Dr. Marcelo Andrade da Costa Vieira
SEL/EESC/USP

APROVADO

Mestre Alex Antonio Affonso
Doutorando - SEL/EESC/USP

APROVADO

Coordenador do Curso Interunidades - Engenharia de Computação:

Prof. Associado Evandro Luís Linhari Rodrigues

AGRADECIMENTOS

Agradeço primeiramente a Deus por sempre me guiar durante todos os momentos em minha vida.

Agradeço aos meus pais por sempre me apoiarem e não medirem esforços para a realização dos meus sonhos.

Agradeço ao Prof. Evandro que me orientou desde o segundo ano na faculdade, foi um grande motivador durante a graduação e sempre esteve disposto a ajudar resolver todos os problemas que surgiram.

Agradeço a todos os professores que estiveram sempre dispostos a ensinar, ainda que de um modo não tão fácil.

RESUMO

Com o avanço dos dispositivos eletrônicos os sistemas embarcados tornaram-se cada vez mais presentes no dia-a-dia e uma série de aplicações surgiram para facilitar tarefas corriqueiras. Com a ajuda dos sistemas operacionais *Linux* expandiram-se as possibilidades de desenvolvimento devido a gama de recursos e ferramentas disponíveis para tal sistema. Dentre as plataformas de desenvolvimento disponíveis, a *Beaglebone Black* é amplamente apoiada pela comunidade *opensource*, possui compatibilidade com uma série de sistemas operacionais, periféricos e bibliotecas. Visando estudar conceitos sobre visão computacional e considerando-se que a identificação facial é um método de identificação seguro e eficaz, este projeto tem por intuito o desenvolvimento de um sistema de identificação facial o qual produz uma lista de presença similar às utilizadas nas salas de aulas. Para tanto, utiliza-se *Linux* embarcado em conjunto com as rotinas disponíveis na biblioteca *OpenCV* e desenvolve-se uma interface Web para o controlar a aplicação. Além disso, realiza-se um estudo sobre algoritmos de detecção de faces e algoritmo de identificação facial presentes na literatura, como também uma análise da eficiência do seu uso na plataforma embarcada. Ao término, coleta-se a porcentagem de acerto de cada algoritmo bem como o tempo médio necessário para identificar uma face.

Palavras-chave: *Sistemas embarcados, Linux embarcado, Beaglebone, OpenCV, reconhecimento facial, visão computacional.*

Abstract

With advancement of electronic devices, the embedded systems have become ever more present in day to day in a series of applications that emerged to facilitate the simple tasks in the human life. Together with Linux operating systems, showed up a series of possibilities of development due the range of resources and tools available for such system. Among the development platforms available, the Beaglebone Black is fully supported by community open source and has compatibility with a lot of operating systems, peripheral devices and libraries. In order to study concepts of computer vision and taking into consideration that facial identification is a safe and effective identification method, this project is meant to develop a facial identification system, which produces a list similar of presence list used in the classroom. Therefore, it uses embedded Linux in conjunction with the routines available in the OpenCV library and develops a Web interface to control the application. In addition, a study is carried out on face detection algorithms and facial recognition algorithm in the literature, as well as an analysis of the efficiency of its use in the embedded platform. At the end, collecting the percentage of accuracy of each algorithm and the average time required to identify a face.

Keywords: **Embedded systems, embedded Linux, Beaglebone, OpenCV, face recognition, computer vision..**

Lista de Figuras

Figura 1 - Diagrama de interfaces de um computador.....	7
Figura 2 - Representação dos classificadores em cascata utilizado no algoritmo de Viola Jones fonte:[12].....	10
Figura 3 - Representação da operação LBP fonte:[13].....	11
Figura 4 - Operador LBP estendido fonte: [13].....	11
Figura 5 - BeagleBone Black principais componentes fonte: [8].....	13
Figura 6 - WebCam Logitech C270.....	14
Figura 7 - Diagrama de estados simplificado	21
Figura 8 - Página web da interface de controle.....	23
Figura 9 - Página web da aba Estático.....	24
Figura 10 - Página web da aba Ao Vivo.....	24
Figura 11 - Página web da aba Cadastro.....	25
Figura 12 - Página web da aba Relatório.....	25
Figura 13 - Página web da aba Controle.....	26
Figura 14 - Página web da aba Dados	26
Figura 15 - Diagrama que representa a comunicação entre interface web/Software em C++	27
Figura 16 - Gráfico de processamento da Beaglebone Black	30
Figura 17 - Gráfico da média de tempos(s) x Threshold do algoritmo que usa Haar Cascade	31
Figura 18- Gráfico da porcentagem de sucesso/falha na identificação em função do Threshold do algoritmo que usa Haar Cascade.....	32
Figura 19 - Gráfico da média de tempos(s) x Threshold do algoritmo de detecção que usa LBP	33
Figura 20- Gráfico da porcentagem de sucesso/falha na identificação em função do Threshold do algoritmo que usa LBP	34
Figura 21 - Relatório de presença gerado automaticamente	34

Abreviaturas

OpenCV (Open Source Computer Vision Library)

CISC (Complex Instruction Set Computer)

RISC (Reduced Instruction Set Computer)

ARM (Advanced RISC Machine)

USB (Universal Serial Bus)

XML (eXtensible Markup Language)

SSH (Secure Shell)

API (Application Program Interface)

MJPG (Motion Joint Photographic Experts Group)

UDP (User Data Protocol)

PHP (Personal Home Page)

HTML (HyperText Markup Language)

FIFO (First In First Out)

LCD (Liquid Crystal Display)

CRT (Cathodic Ray Tube)

GUI (Graphical User Interface)

Sumário

1. Introdução	1
1.1. Objetivos gerais	2
1.2. Justificativa	2
1.3. Organização do trabalho.....	4
2. Fundamentação Teórica.....	5
2.1. <i>Sistemas embarcados</i>	5
2.2. <i>Sistema Operacional</i>	6
2.3. <i>Linux</i>	7
2.4. <i>Algoritmos de detecção de faces</i>	8
2.4.1. <i>Algoritmo de Viola Jones</i>	9
2.4.2. <i>Algoritmo Padrão Binário Local (Local Binary Patterns)</i>	10
2.5. <i>Autofaces (Eigenfaces)</i>	11
3. Materiais utilizados	13
3.1. <i>Beaglebone Black</i>	13
3.2. <i>WebCam 720p</i>	14
4. Desenvolvimento e implementação.....	15
4.1. <i>Configuração e Instalação</i>	15
4.2. <i>Biblioteca OpenCV</i>	16
4.3. <i>Servidor Web Apache</i>	17
4.4. <i>Mjpg-Streamer</i>	18
4.5. <i>Software em C++</i>	20
4.6. <i>Interface Web de controle</i>	23
4.7. <i>Integração da Interface Web de controle e do Software em C++</i>	27
5. Resultados	29
5.1. <i>Análise de processamento</i>	30
5.2. <i>Análise algoritmo de detecção de Viola Jones</i>	31
5.3. <i>Análise do algoritmo de detecção com LBP</i>	33
5.4. <i>Relatório de presença</i>	34
6. Conclusões	35
Referências bibliográficas.....	37
APÊNDICE A - <i>Instalação e configuração do Ubuntu 14 LTS na BBB</i>	41
APÊNDICE B - <i>Código Software em C++</i>	43
APÊNDICE C - <i>Tabela dados do processamento da BeagleBone Black</i>	53
APÊNDICE D - <i>Tabelas de testes do algoritmo de detecção com Haar Cascade</i>	55
APÊNDICE E - <i>Tabelas testes do algoritmo de detecção com LBP</i>	61

1. Introdução

Muito se debate, hoje em dia, sobre segurança de sistemas de controle de acesso. Os métodos empregados largamente até então tais como documentos de identidade e senhas de autenticação, não se mostram mais eficientes no cenário atual onde documentos são falsificados e senhas são roubadas de maneira muito simples. A demanda por segurança e facilidade na usabilidade do usuário fez-se uma excelente oportunidade para utilizar um sistema de identificação e ou autenticação não dependente de características que podem ser roubadas ou deturpadas facilmente. Dessa forma, a utilização de características biométricas como a face humana para valer-se como mecanismo de identificação é de grande valia, já que a coleta de amostras é simples e não requer nenhum dispositivo especial para captura e além de ser mais eficiente e segura se comparadas com a utilização dos métodos tradicionais, tais como documentos de identidade, cartões eletromagnéticos etc.

Com o avanço da microeletrônica nas últimas décadas a escala dos dispositivos reduziu-se drasticamente e hoje tem-se microcomputadores do tamanho de um *flash drive stick* permitindo o desenvolvimento de produtos de baixo custo. Sendo assim, surgiram uma série de aplicações em sistemas embarcados [1] que tem por intuito facilitar e otimizar o tempo e a eficiência de algumas tarefas do dia-a-dia. Tal propósito condiz com a finalidade de um sistema embarcado, o qual por definição é um sistema em escala reduzia e dedicado a executar uma tarefa exclusiva.

Portanto, a proposta deste projeto é utilizar uma placa de baixo custo, *Beaglebone Black* [2] e uma *webcam* 720p de uso comum, em conjunto com o sistema operacional *Linux* embarcado para realizar a detecção e identificação de faces de pessoas, permitindo então controlar ou registrar o acesso de indivíduos em um meio. Dessa forma, tem-se um produto de baixa complexidade, baixo valor agregado, portátil, eficiente, seguro e rápido. Otimizando-se, por exemplo, a tarefa de um professor de realizar a chamada oral em uma sala de aula.

1.1. Objetivos gerais

Dentre as finalidades almejadas deste projeto, busca-se aprofundar o conhecimento sobre o desenvolvimento de aplicações para sistemas embarcados aplicados em processamento de imagens e vídeos e a utilização de conceitos de visão computacional, bem como a análise da performance de alguns algoritmos de detecção de faces.

Propõe-se, assim, os seguintes objetivos específicos para o projeto:

- ❖ Utilização de *Linux* Embarcado.
- ❖ Utilização de *hardware* de baixo custo e baixa complexidade.
- ❖ Utilização da biblioteca *OpenCV* (*Open Source Computer Vision Library*) [3].
- ❖ Técnicas para tratamento de multimídia (imagens/vídeo).
- ❖ Estudos sobre Visão computacional.
- ❖ Análise e comparação dos algoritmos detecção e identificação de faces humanas.

1.2. Justificativa

O desenvolvimento de um sistema de identificação facial utilizando uma plataforma embarcada de baixo custo e uma *webcam* de uso comum, é uma alternativa viável para realizar o controle e ou registro de acesso de pessoas em um determinado meio, pois permite a portabilidade/mobilidade e não exige gastos elevados para montar a solução. Além disso, tal projeto viabiliza aprofundar os conhecimentos em *Linux* embarcado [4] bem como a utilização de ferramentas para o tratamento de imagens e de vídeos, como a biblioteca *OpenCV*. O completo desenvolvimento da plataforma ainda envolve sólidos conhecimentos de programação em linguagem C/C++, além da configuração e instalação do servidor *Web*, o desenvolvimento de uma página em *PHP* para visualização dos resultados, bem como a integração da página *Web* e da aplicação C++, como será explanado posteriormente.

Em suma, o sistema desenvolvido neste projeto, tem por intuito reproduzir a lista de presença de pessoas em um determinado local, neste exemplo uma sala de aula, utilizando o reconhecimento facial como meio de identificação. Para tanto, foi necessário desenvolver uma plataforma *Web* que atue como interface de controle entre o usuário e o *software* em C++ que dentre outras coisas descritas a seguir, executa o processamento de imagens.

A interface web de controle é responsável por gerenciar o sistema, dessa forma utilizando-a é possível controlar a *BeagleBone Black* (reiniciar, fechar, inicializar e ver status da aplicação), cadastrar e remover novos usuários utilizando-se o nome, o sobrenome e o número USP (ex: 7126883). Cadastrar e remover novos relatórios de presença utilizando-se nome da disciplina, turma, professor responsável, hora de início e hora de término. Visualizar o vídeo ao vivo da *WebCam*, visualizar a última pessoa identificada, visualizar usuários e relatórios cadastrados.

Concomitantemente, um *software* em C++ é executado no sistema *Linux* e este recebe instruções da interface web de controle via uma fila compartilhada (uma área da memória do Linux alocada como uma *FIFO*) a qual é acessada por ambas aplicações.

Basicamente, este *software* opera como uma máquina de estados, onde cada estado representa um comportamento da aplicação, como por exemplo coletar fotos, treinar banco de imagens, reconhecer fotos etc. A mudança de estados ocorre por meio das instruções provenientes da interface de controle. Este processo, será descrito mais detalhadamente na seção 4.7.

1.3. Organização do trabalho

Tal projeto foi subdivido em seis capítulos, sendo estes dispostos da seguinte maneira:

No primeiro capítulo, este em questão, apresenta-se a ideia inicial que culminou com a realização do projeto, assim como os objetivos almejados com este desenvolvimento.

Já no segundo capítulo, introduz-se a fundamentação teórica necessária para um completo entendimento sobre o projeto.

Dando continuidade, no terceiro capítulo apresenta-se a descrição geral sobre o projeto e dos materiais utilizados.

No quarto capítulo ocorre a explanação sobre desenvolvimento e implementação de cada etapa do projeto.

A apresentação, análise e discussão dos resultados são encontradas no quinto capítulo.

Ao final, no sexto capítulo, apresenta-se a conclusão do projeto bem como possíveis projetos futuros e ou alterações deste trabalho.

2. Fundamentação Teórica

Para o desenvolvimento deste sistema foram adotadas algumas tecnologias cada qual com suas particularidades. Nas seções subsequentes serão apresentados fundamentos teóricos em cada um dos elementos utilizados durante o desenvolvimento do projeto bem como suas vantagens pelas quais foram adotados.

2.1. Sistemas embarcados

É difundido e reconhecido que o uso dos computadores é encontrado no processamento de informações para a humanidade. Seja para escrever um livro, buscar informação no *Google*, comunicar-se via e-mail ou acessar as redes sociais e uma série de outras aplicações nas quais o computador é indispensável. Todavia, uma série de computadores são utilizados em várias outras aplicações que não são tão claras e difundidas para o usuário comum. Dentre elas, cita-se que há computadores controlando os freios, cintos, *airbags* e sistemas de áudio dos carros, controlando o micro-ondas, geladeiras e todos eletrodomésticos encontrados nas casas. Eles codificam/decodificam e enviam o sinal de voz e/ou vídeo dos celulares. Eles controlam os robôs das fábricas, os semáforos das ruas, constroem e processam imagens do corpo humano para estudos biomédicos. Controlam metrôs, trens, navios e aviões. Em suma, fazem parte da vida humana muito além do que se imagina. Estes computadores “invisíveis” são conhecidos como sistemas embarcados [4] e o software que é executado em tais sistemas é conhecido como software embarcado.

Enquanto os sistemas de propósito geral, como um computador *desktop*, realiza diversas tarefas, e possui recursos para processar aplicações distintas, inclusive ao mesmo tempo, os sistemas embarcados são projetados com recursos limitados de processamento e de energia. Portanto, são idealizados para executar tarefas dedicadas e específicas, muitas vezes com requisitos específicos de tempo ou energia, como no caso de sistemas críticos os quais controlam os freios e *airbags* dos carros que devem responder em tempo real e os robôs que operam em áreas remotas que devem requerer o mínimo de energia necessária, como no caso do robô que está em marte [5].

Os sistemas embarcados são apresentados em diversas arquiteturas, sendo estas definidas pelo microcontrolador [6] utilizado. Dentre as opções encontradas, evidencia-se o INTEL 8051[7] o qual foi lançado em 1977 e pertence à família *MSC-51*, possui 8 *bits* e é tido

como um dos mais populares do mundo pelo fato de possuir um conjunto de instruções C/SC o que lhe oferece um vasto conjunto de instruções. Também é necessário mencionar os microcontroladores da *MICROCHIP*[8] que possuem uma grande gama de dispositivos de famílias de 8 *bits* até 32 *bits*.

Além destes, tem-se os microcontroladores *ARM* [9] os quais são amplamente utilizados em celulares, *tablets*, *drones* e *dongles* etc. É uma arquitetura de 32 *bits* que utiliza o conjunto de instruções R/SC a qual possui reduzido número de instruções se comparada com a arquitetura C/SC, logo reduz os custos e o consumo de energia, sendo estas as principais vantagens dos microcontroladores *ARM*.

2.2. Sistema Operacional

Um sistema computacional, seja embarcado ou de propósito geral, é composto por uma série de dispositivos dentre os quais pode-se ter o processador, memória, dispositivo de armazenamento, mouse, teclado, e alguns periféricos que podem ser embarcados (no caso de sistema ser embarcado) ou conectados externamente via porta *USB*¹. Para cada um destes componentes há diversas características particulares, padrões e protocolos utilizados que diferem de um para outro. Dessa forma, seria oneroso ou inviável desenvolver aplicações específicas para controlar cada dispositivo em cada projeto que os utilizassem. Sendo assim, é de grande valia ter um sistema para controlar e gerenciar tais dispositivos de uma forma transparente para o usuário de forma que a troca de um dispositivo não acarrete no desenvolvimento de um novo código para operar tal dispositivo.

Portanto, uma função primordial de um sistema operacional é trabalhar na interface entre aplicação do usuário e *hardware*, de tal forma que o *hardware* seja transparente para uma aplicação em nível de usuário e essa apenas envie comandos para o sistema operacional, sendo este último o responsável pela entrega da informação ao dispositivo correto e na formatação correta para que o mesmo entenda e execute sua tarefa. Tal situação é evidenciada na Figura 1.

Além de fazer com que a aplicação não se preocupe com o *hardware* que será utilizado, o sistema operacional também é responsável por controlar os recursos do sistema. Quando uma aplicação está sendo executada ela utiliza parte do processamento do processador, da memória e talvez até mesmo do dispositivo de armazenamento. Ao executar

¹ é um tipo de entrada que permite a conexão de periféricos sem a necessidade de desligar o computador.

mais de uma aplicação pode haver concorrência por recursos e há a necessidade de um gerenciamento dos mesmos.

Sendo assim, o sistema operacional é responsável por estabelecer métodos e políticas para distribuir para cada aplicação uma parcela da memória, o tempo que o processador irá executar esta aplicação bem como controlar o acesso aos periféricos. Tal gerenciamento pode tornar-se complexo e é indispensável que a aplicação não se atenha nisto e dedique-se em executar sua tarefa, cabe assim ao sistema operacional gerenciar tais recursos.

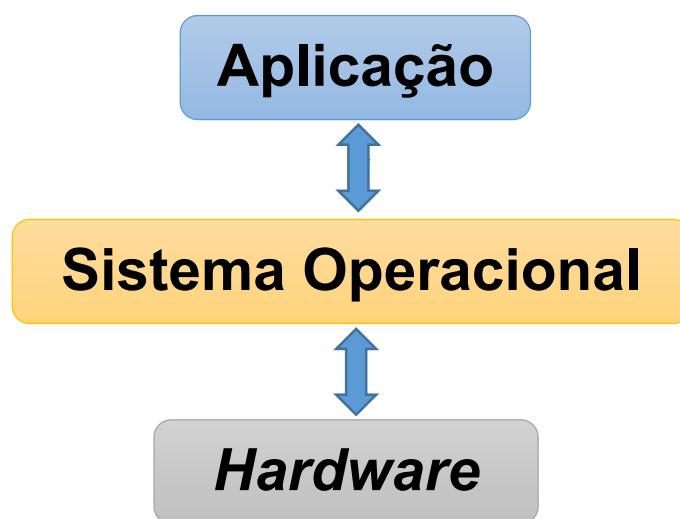


Figura 1 - Diagrama de interfaces de um computador

2.3. Linux

O sistema operacional *Linux* mudou os paradigmas de sistemas operacionais proprietários conhecidos até então[10]. De um lado, os sistemas operacionais proprietários apresentam um ambiente fixo onde as aplicações são específicas e não há grande flexibilidade para customizá-lo, aumentar sua performance ou corrigir eventuais *bugs*². Os sistemas operacionais *Linux* e outros sistemas *open sources* mudaram isto com a possibilidade de customização e redistribuição com acesso completo ao código fonte.

² Um **bug** (termo da língua inglesa que significa, neste contexto, "defeito") é um erro no funcionamento comum de um *software* (ou também de *hardware*), também chamado de **falha** na lógica de um programa, e pode causar comportamentos inesperados, como resultado incorreto ou comportamento indesejado.

Boa parte do sucesso do *Linux* provém do fato de ser código aberto e disponibilizá-lo para toda a comunidade [10]. Isso fomentou a formação de uma comunidade ativa de programadores que melhoram o sistema, reportam *bugs*, implementam novas funcionalidades e disponibilizam gratuitamente na *internet*. Dessa forma, uma quantidade grande de pessoas estão utilizando e melhorando o sistema *Linux*.

Em suma, o sistema operacional *Linux* tornou-se popular devido a sua eficiência, qualidade e gratuidade. Vem sendo utilizado amplamente desde em supercomputadores até mesmo em sistemas embarcados.

Vale ressaltar que o termo *Linux* refere-se originalmente ao *kernel* criado por Linus Torvalds, que roda nos sistemas operacionais *GNU/Linux*. O *kernel* como o próprio nome diz, é um núcleo de ferramentas responsável pelas atividades básicas do sistema operacional *GNU/Linux*, dentre as quais tem-se: gerenciamento de recursos, abstração para aplicação do usuário, gerenciar o sistema de arquivos etc.

Entretanto, o *kernel* sozinho não provê muita utilidade para o usuário, não se pode ter um terminal para comandos básicos, realizar edição de arquivos ou comunicar-se via rede. Para tais tarefas são necessários outros aplicativos tais como terminais, compiladores, editores e bibliotecas de linguagens de programação como C/C++ que não estão inclusos no *kernel*.

Em meados de 1984 surgiu uma coletânea de softwares, bibliotecas, ferramentas de desenvolvimento e até mesmo jogos, denominado projeto *GNU* [11]. Posteriormente houve a combinação do projeto *GNU* com o *kernel* de Linus Torvalds formando um sistema operacional poderoso também chamado sistema operacional *GNU/Linux*, o qual muitos o chamam somente de *Linux*. Neste projeto também será empregado somente o termo *Linux* para referenciar o sistema operacional *GNU/Linux*.

2.4. Algoritmos de detecção de faces

Reconhecimento facial é uma tarefa simples para os humanos que até mesmo um bebê com poucos dias de vida já consegue distinguir entre rostos de conhecidos e de desconhecidos. Enquanto isso, para os computadores este é um processo bem mais complexo e ainda não existe um sistema que se compare a eficiência do cérebro humano.

A detecção de faces é uma das principais tarefas executadas durante uma série de outras aplicações como rastreamento de faces, identificação de faces e remoção de olho vermelho [12], pois antes de qualquer um destes procedimentos citados é necessário delimitar a área da imagem na qual encontra-se a face para então dar sequência em um processamento para identificação, por exemplo. Dessa forma, a utilização de um algoritmo eficiente para detectar a face antes de realizar a identificação facial, por exemplo, implicará diretamente na eficiência da aplicação de identificação, já que um erro na primeira etapa culmina com um erro no reconhecimento.

A princípio não parece muito complexo, porém uma série de desafios estão associados a este processo. Dentre os fatores [12] que podem dificultar o procedimento, tem-se:

- A pose da pessoa na foto (frontal, perfil etc.)
- Presença ou não de barba, bigode, óculos.
- Expressões faciais.
- Orientação da câmera.
- Condições do ambiente (luz, brilho etc.)

Existem diversos algoritmos utilizados para a detecção de objetos, dentre os quais optou-se por utilizar o algoritmo de Viola Jones muito utilizado para detectar faces e o algoritmo de Local Binary Pattern (LBP), utilizado para identificação de micro padrões, muito utilizado para detecção de textura e que foi utilizado neste projeto para detecção de faces.

2.4.1. Algoritmo de Viola Jones

Nos últimos anos, obteve-se muito progresso em técnicas [13] para detecção facial. Dentre as pesquisas realizadas, o detector de objetos Viola-Jones [14] conseguiu grande destaque devido a sua eficiência em detectar faces com elevada taxa de acertos, baixa taxa de falso positivo e principalmente devido ao seu baixo custo computacional [15].

Subdividido em três partes, o algoritmo de Viola-Jones primeiramente utiliza filtros de *Haar* para representar a imagem em um espaço característico. Isto é realizado utilizando-se a “Imagem Integral” [14] a qual é responsável por aumentar o ganho computacional pois as operações são realizadas sobre retângulos ao invés de operar sobre *pixels*. Em um segundo passo, utiliza-se um classificador baseado método *AdaBoosting* [14] responsável por selecionar as características mais relevantes extraídas no passo anterior. Para obter um

melhor desempenho utiliza-se classificadores em cascata. A Figura 2 ilustra o funcionamento dos classificadores em cascata.

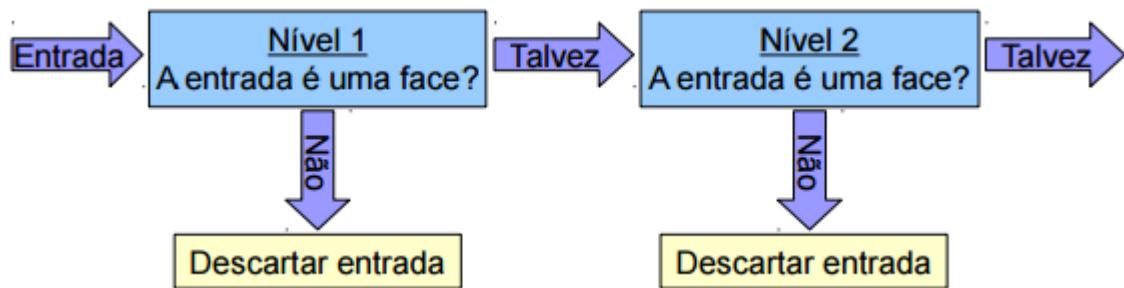


Figura 2 - Representação dos classificadores em cascata utilizado no algoritmo de Viola Jones

fonte:[12]

Para realizar a detecção utiliza-se as características selecionadas no passo anterior, as quais são escalonadas em diversos tamanhos já que não se sabe o tamanho nem a posição da face na imagem alvo, e então são aplicadas em todas sub janelas da imagem para verificar se a característica está presente ou não na imagem. Há uma série de classificadores já definidos e salvos em arquivos *XML* que podem ser utilizados diretamente por ferramentas como *OpenCV*, como será mostrado posteriormente.

2.4.2. Algoritmo Padrão Binário Local (*Local Binary Patterns*)

Além do algoritmo de Viola Jones utilizou-se o algoritmo de *Local Binary Patterns* (*LBP*), o qual também utiliza o algoritmo *Adaboost* para realizar o treinamento dos classificadores [16].

A utilização do *LBP* em uma imagem permite resumir a estrutura espacial de uma pequena parcela da imagem (8 *pixels*) em um único número (código *LBP*). Tal código é definido a partir de uma vizinhança de 3x3 *pixels*, comparando-se os pixels externos com o *pixel* central.

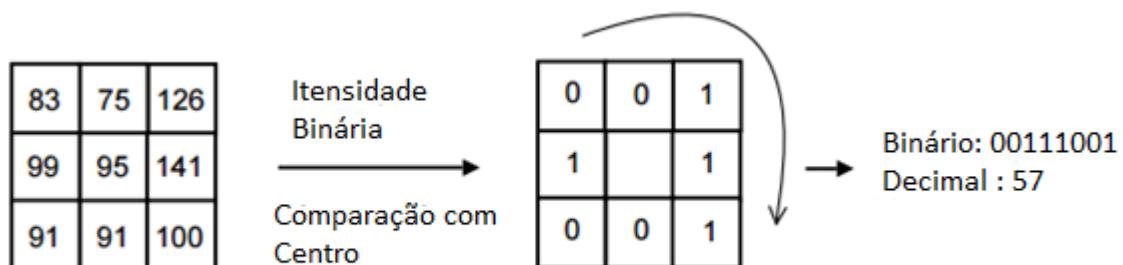
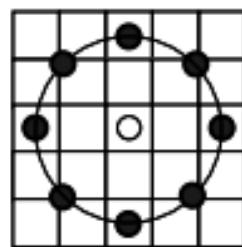


Figura 3 - Representação da operação *LBP*

fonte:[13]

A *Figura 3* ilustra a operação para obter o código *LBP* onde observa-se que se a subtração dos pixels das bordas do pixel central (95) for maior ou igual a 0 atribui-se o valor 1 e se for menor que 0 atribui-se o valor 0. Ao final, tem-se o número binário 57 que representa um micro padrão da estrutura.

Posteriormente, o operador *LBP* foi estendido [17] para utilizar círculos de diferentes raios para definir o código *LBP*, conforme observa-se na *Figura 4* que representa o $LBP_{8,2}$ (8 pontos e raio 2). Os pontos que não estão no centro dos pixels são definidos por interpolação.

Figura 4 - Operador *LBP* estendido

fonte: [13]

Uma face de uma imagem pode ser considerada uma combinação de micro padrões que podem ser identificados via *LBP*. Portanto, treinando classificadores com o algoritmo *AdaBoost*, pode-se definir um detector *LBP* que pode ser utilizado para detectar faces.

2.5. *Autofaces (Eigenfaces)*

Basicamente, utilizar um sistema computacional para realizar o reconhecimento facial consiste em detectar características do indivíduo, tais como olhos, nariz, contorno do rosto e definir um modelo de face de acordo com as posições, tamanhos e relações entre as características [18].

Um conjunto de *Eigenfaces* ou “*face space*” é o termo denominado para o conjunto de autovetores³, quando estes são utilizados no campo de visão computacional. Tal conjunto representa a coleção de características de um grupo de imagens [18].

Para obter-se um conjunto de autofaces realiza-se sobre o banco de dados de imagens a análise dos componentes principais (*PCA – Principal component analysis*). Com tal conjunto constrói-se o que é chamado de “*face space*”, um subespaço onde serão projetadas as imagens de teste e estas serão classificadas de acordo com a sua posição em relação às imagens conhecidas previamente. Dessa forma, dentro de algumas limitações o reconhecimento facial é alcançado.

³ Em Álgebra linear, um autovetor ou vetor próprio representa uma direção que é preservada por uma transformação linear

3. Materiais utilizados

Neste capítulo serão apresentadas as principais especificações técnicas dos materiais utilizados neste projeto.

3.1. Beaglebone Black

A solução de hardware escolhida para execução deste projeto foi a *BeagleBone Black (BBB)*, encontrada na Figura 5. Tal plataforma é da família *BeagleBoard*, apresenta baixo custo e é baseada no *Sitara XAM3359AZCZ100*, processador ARM Cortex A8 da *Texas Instruments*.

Dentre as principais características, ressalta-se:

- Processador AM3358 1GHz
 - DDR3 512MB / 800MHz
 - 4GB eMMC (*Flash*)
 - Saída *HDMI*
 - *Debug Serial Header*
 - *USB 2.0 e Ethernet*
 - *2x 46 pinos*
 - Compatibilidade com: *Android, Ubuntu, Debian, Fedora, QNX etc.*

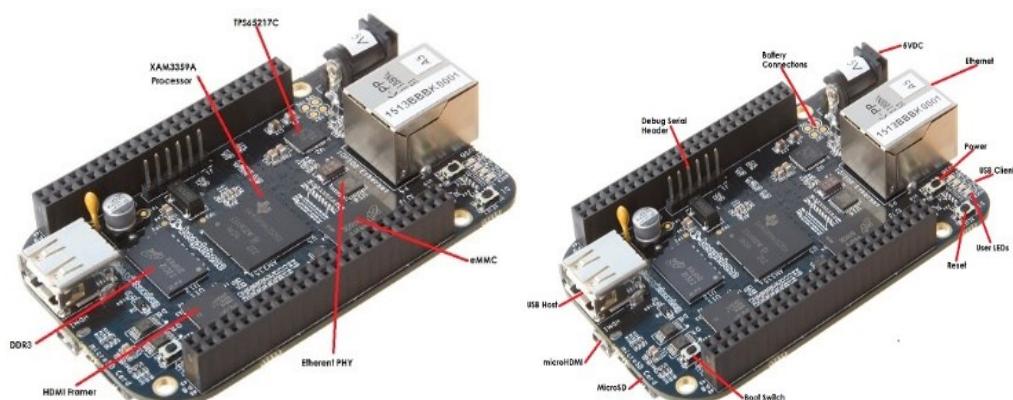


Figura 5 - *BeagleBone Black* principais componentes
fonte: [8]

Amplamente amparada pela comunidade *open source*, a *BBB* é uma excelente alternativa para quem busca rápida prototipagem, baixo custo, estabilidade de desenvolvimento e uma gama de recursos e aplicações disponíveis na *internet*. Além disso,

possui a possibilidade de utilizar outros periféricos, como display *LCDs*, sensores, antenas, webcams etc.

3.2. *WebCam 720p*

Para realizar a captura de imagens e de vídeos utilizou-se uma webcam de uso comum, com resolução de até 1080x720p da marca *Logitech*, modelo C270, representada na *Figura 6*.

Especificações técnicas:

- Videochamada HD (1280 x 720 pixels)
- Captura de vídeo: até 1280 x 720 pixels
- Tecnologia *Logitech Fluid Crystal™*
- Fotos: até 3,0 *mexapixels* (com otimização do *software*)
- Microfone embutido com redução de ruídos
- *USB 2.0* de alta velocidade certificado (recomendado)
- Clipe universal que se ajusta a laptops e monitores *LCD* ou *CRT*



Figura 6 - *WebCam Logitech C270*

4. Desenvolvimento e implementação

Dentre as possibilidades, adotou-se o modelo de desenvolvimento de *software* incremental. Ainda que não tão rígido, tentou-se dividir a solução em pequenos módulos, cada qual desenvolvido paralelamente e incrementalmente. Ao final, integrou-se todas as partes para compor a solução final.

Portanto, o desenvolvimento deste projeto baseou-se em cinco etapas principais. Instalação do sistema operacional *Linux Ubuntu*, a compilação e a instalação da biblioteca *OpenCV* para linguagem C/C++, instalação do servidor *Web Apache*, compilação e instalação da ferramenta *mjpg-streamer* e desenvolvimento da aplicação em linguagem C++ e da página *Web*. Nas seções subsequentes serão explanados com maiores detalhes de como ocorreu cada etapa, os passos durante a instalação que foram encontrados após várias tentativas, bem como os problemas enfrentados.

4.1. Configuração e Instalação

É de suma importância a utilização de um sistema operacional em uma plataforma embarcada pois, conforme explanado na seção 2.2, isto torna o hardware transparente à aplicação em nível de usuário e permite que mais de uma aplicação seja executada ao mesmo tempo, bem como a interação entre elas.

A *BeagleBone Black* é distribuída com o sistema operacional *Angström* [19], todavia, devido a abundância de recursos para a distribuição *Ubuntu* e alta estabilidade deste sistema na BBB, optou-se por utilizar uma imagem pré-compilada do *Ubuntu 14.04 LTS* com *kernel* 3.8.13-bone63 a qual foi gravada na memória *flash* da *Beagle*, dispensando a necessidade de utilizar o botão de boot para realizar o procedimento de boot. Sendo assim, a todo instante que a BBB é conectada à alimentação de energia o *boot* será realizado automaticamente.

Para a conclusão desta etapa foram realizadas uma série de pesquisas para encontrar a melhor imagem e o melhor modo de realizar o a escrita na memória *flash*, e após inúmeras tentativas chegou-se ao procedimento encontrado no Apêndice A, que descreve em detalhes como instalar o *Ubuntu 14.04 LTS* na memória *flash* da BBB.

4.2. Biblioteca OpenCV

OpenCV é uma biblioteca de licença *BSD* de código aberto, disponível em C, C++, Java, Python, MacOS, IOS e android que inclui centenas de algoritmos de visão computacional e possui suporte a aceleradores de hardware.

Dentre os algoritmos utilizados neste projeto tem-se para detecção de faces o algoritmo Viola Jones com classificador *haar cascade* tipo frontal e algoritmo *LBP*. Para o reconhecimento facial o algoritmo baseado em *Autofaces*. Tais algoritmos foram escolhidos devido ao fato de a biblioteca ter a *API* disponível para utilizá-los, assim como os classificadores já treinados e também foram os algoritmos encontrados na literatura já utilizados com tal biblioteca.

A instalação da *OpenCV* demanda bastante tempo durante o desenvolvimento, pois os parâmetros que são usualmente utilizados na linha de comando em uma compilação *desktop* não são aceitos, em alguns casos, na plataforma embarcada, logo não foi um processo trivial como parece seguindo-se os tutoriais encontrados na internet. Além disso, foi necessário encontrar um *toolchain*⁴ que realizasse a compilação correta na BBB. Em suma, o processo total de compilação da *OpenCV* na Beagle dispensou em torno de 9h, e várias vezes não chegou a conclusão devido a um parâmetro não suportado ou à falta de alguma biblioteca. Após muita pesquisa e muitos testes, chegou-se ao procedimento abaixo que compila e instala a *OpenCV* na *beaglebone black*.

Primeiramente, instala-se os pacotes essenciais para compilá-la, por meio do comando:

```
apt-get install build-essential cmake
```

Faz-se o *download* do código fonte.

```
wget http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.9/opencv-2.4.9.zip
```

Descompacta-se e cria-se uma pasta chamada “*build*” dentro do diretório “*opencv-2.4.9*”

A partir deste diretório utiliza-se o comando abaixo para configurar a compilação:

⁴ É um conjunto de ferramentas necessárias para a compilação.

```
cmake -DSOFTFP=ON -DCMAKE_TOOLCHAIN_FILE=../platforms/linux/arm-gnueabi.toolchain.cmake ..
```

E então chama-se a ferramenta `Make` e em seguida `Make Install` para compilar e instalar as bibliotecas que por padrão vão para um diretório chamado “*Install*”. Conforme observa-se na linha de comando anterior não foram definidos quase nenhum parâmetro extra, e sim utiliza-se somente os parâmetros estabelecidos por padrão, além de utilizar um toolchain que gere um binário para arquitetura ARM.

Dentro da pasta “*Install*” encontra-se as pastas “lib”, “include”, “share” e “bin” as quais contém os arquivos necessários para utilização da biblioteca *OpenCV*. Tais pastas devem ser copiadas para a pasta “/usr”. Posteriormente, deve-se editar o arquivo “/usr/lib/pkgconfig/opencv.pc” e mudar a variável *prefix* para “prefix=/usr”, desse modo o Linux procurará as bibliotecas da *OpenCV* nesta pasta.

Utilizando-se o comando abaixo, o *Linux* encontra as bibliotecas instaladas.

```
pkg-config --cflags opencv
pkg-config --libs opencv
```

4.3. Servidor Web Apache

Conforme descrito anteriormente, utilizou-se uma interface *Web* para gerenciar a aplicação e visualizar alguns resultados gráficos, como imagens e vídeo. Para tanto, optou-se por utilizar o servidor web *Apache* [20] o qual tem amplo material disponível na internet, é de fácil configuração e instalação na *BBB*.

Para instalá-lo, utiliza-se:

```
apt-get install apache2
```

Após a instalação será criado um diretório na pasta “/usr/www/html” o qual deve conter todo os arquivos da página que serão acessados via *browser* através do *IP* real da *Beagle*.

4.4. Mjpg-Streamer

MJPEG-Streamer [21] é uma aplicação desenvolvida em C a qual consegue realizar *streaming* de vídeo em *MJPEG* via *HTTP* para uma página *Web*. Basicamente, esta ferramenta funciona por meio de *plugins* e permite que um *plugin* de entrada gere uma saída para um *plugin* de saída. Neste projeto utilizou-se três *plugins*: *input_uvc*, *output_file*, *output_http*. Sendo responsáveis pela aquisição do vídeo da *webcam*, escrita das imagens em um diretório e envio do vídeo para a página *HTML*, respectivamente.

A instalação segue as linhas de comando abaixo:

Instalando-se os pré-requisitos:

```
sudo apt-get install g++ curl pkg-config libv4l-dev libjpeg-dev build-essential libssl-dev vim
cmake
sudo apt-get install imagemagick
```

Realizando-se o *download*:

```
wget https://github.com/shrkey/mjpg-streamer/raw/master/mjpg-streamer.tar.gz
tar -xvf ./mjpg-streamer.tar.gz
```

Compilando-se:

```
cd mjpg-streamer
make
sudo make install
```

Após a instalação é possível encontrar na pasta *mjpg-streamer* os *plugins* compilados bem como a pasta “www” a qual contém os arquivos que serão exibidos no *browser*, porém o seu acesso deve ser feito por meio de uma porta específica, por padrão o *mjpg-streamer* utiliza a porta 8080. Dessa forma o acesso deve ser feito via <http://192.168.7.2:8080>, via de exemplo.

A execução e utilização do script podem ser feitas de diversas formas, pois há várias combinações possíveis. Há um arquivo dentro da pasta *mjpg-streamer* que contém vários exemplos de utilização, chamando “start.sh”.

Como exemplo, segue uma das possibilidades:

```
./mjpg_streamer -i "./input_uvc.so" -o "./output_http.so -w ./www"
```

A sintaxe de argumentos utilizados no *mjpg-streamer* é simples, após a *flag* “ -i ” insere-se a linha de comando referente ao *plugin* de entrada, e após a *flag* “ -o ” insere-se a linha de comando referente ao *plugin* de saída.

Viu-se a necessidade de utilizar o *MJPG-Streamer* para disponibilizar o vídeo ao vivo proveniente da *webcam* de forma simples e com baixo custo computacional. Apesar de a biblioteca *OpenCV* ter rotinas para capturar o vídeo, não é trivial codificar este vídeo e disponibilizá-lo em uma página Web. Dentre as alternativas, pensou-se em gerar uma *stream UDP* e recebê-la na página local por meio de algum *plugin* com suporte a execução de tal *stream*, todavia isso exigiria recursos da *BBB* e como será visto na sequência, a aplicação que processa as imagens e realiza o reconhecimento facial já está consumindo quase que a totalidade de recursos da plataforma.

Além disso, como já mencionado, o *MJPG Streamer* possui sua própria pasta *www* a qual irá exibir arquivos *HTML* através de rotinas que simulam requisições *HTTP* na porta 8080. Contudo, em tal pasta não é permitido a execução de scripts *PHP* e devido ao fato de a interface de controle *web* exigir scripts *PHP*, utiliza-se o servidor *Apache* o qual já possui inúmeros recursos, inclusive suporte ao módulo *php5*.

Entretanto, para disponibilizar o vídeo *online* ainda será necessário utilizar o *link* que dá acesso ao vídeo gerado pelo *MJPG Streamer* (<http://192.168.7.2:8080/?action=stream>), o qual pode ser inserido em outra página *HTML* ou *PHP* presente no servidor *Apache*.

Por fim, a linha de comando abaixo gera o vídeo ao vivo no *link* acima, além de escrever, a cada 0,5s as imagens da câmera na pasta “/home/ubuntu/mjpg-streamer”. Tais imagens serão utilizadas no *software em C++* o qual realizará o reconhecimento facial.

```
/home/ubuntu/mjpg-streamer/mjpg_streamer -i "/home/ubuntu/mjpg-streamer/plugins/input_uvc/input_uvc.so" -f 15" -o "/home/ubuntu/mjpg-streamer/plugins/output_http/output_http.so" -w /home/ubuntu/mjpg-streamer/www" -o "/home/ubuntu/mjpg-streamer/plugins/output_file/output_file.so" -s 3 -f /home/ubuntu/mjpg-streamer -d 500" &
```

4.5. Software em C++

Baseando-se no trabalho de Shervin [22][23], o qual desenvolveu uma aplicação em C++ utilizando a biblioteca *OpenCV* para *Linux desktop* onde realiza-se a detecção, reconhecimento facial e disponibiliza os resultados em uma interface gráfica da própria *OpenCV*. O software desenvolvido neste projeto tem como alvo uma plataforma embarcada sem a utilização de interface gráfica e com a utilização de uma interface *Web* de controle. Sendo assim, teve-se como base o código disponibilizado em [23] com uma série de alterações para funcionar no *Linux* embarcado da *Beaglebone Black* bem como para interagir com a interface *web* de controle, conforme pode ser observado no apêndice B.

Primeiramente, o modelo adotado para esta aplicação assemelha-se à uma máquina de estados a qual possui os seguintes estados, sendo o primeiro nome o utilizado no código fonte e o segundo a sua tradução.

START_UP / ÍNICO: Estado inicial da aplicação onde são carregados na memória os dados referentes às imagens e aos relatórios, que estão salvos em arquivos *XML* na memória *flash*.

DETECTION / DETECÇÃO: Estado responsável pela detecção de faces, que consiste em ler imagens provenientes do *mjpg-streamer* e realizar a detecção de faces utilizando algoritmos explanados na seção 2.4.

COLLECT_FACES / COLETA DE FACES: Estado responsável pela coleta de faces para compor o banco de imagens. Para cada novo usuário inserido no sistema são coletadas 25 fotos frontais as quais também são espelhadas. A técnica do espelhamento de imagens é utilizada para aumentar o tamanho do banco de imagens de cada indivíduo bem como para torná-lo mais simétrico. A utilização de 25 imagens foi definida para aumentar a precisão da identificação, pois quanto mais imagens forem utilizadas no banco de imagens melhor será a resposta da aplicação. Todavia, um número muito grande torna a aplicação mais lenta, pois o banco de dados é mantido em memória. Portanto, cada indivíduo contribui com 50 imagens para compor o banco de imagens.

TRAINING / TREINAMENTO: Estado responsável por gerar o conjunto de *Eigenfaces* que será utilizado no reconhecimento facial. Após o treinamento é gerado o banco de imagens, que além de ser salvo na memória é salvo em um arquivo *XML*. A *OpenCV* possui uma API completa para salvar estruturas de imagens de forma que fique fácil recuperá-las para a memória.

RECOGNITION / RECONHECIMENTO: Estado responsável por utilizar o modelo treinado no estado anterior, bem como as imagens provenientes do estado *detection* e realizar o reconhecimento facial.

IDLE / OCIOSO: Como o próprio nome diz, este estado simplesmente coloca a aplicação em modo ocioso, e é utilizado quando faz-se alguma alteração como inserção/remoção de novo usuário ou relatório.

EXIT / SAÍDA: Estado responsável por fechar a aplicação.

A Figura 7, representa o diagrama simplificado de estados da aplicação. Vale ressaltar que os estados OC/OSO e SAÍDA foram omitidos, pois estes são acessados isoladamente somente quando a interface de controle Web envia instruções para executá-los, e não são acessados durante o fluxo de controle da aplicação.

Além disso, não foram inseridas as entradas que desencadeiam as transições de estados, pois para tanto, uma série de rotinas observadas no apêndice B devem ser executadas. Porém, conforme a indicação das flechas pode-se inferir o fluxo de dados entre cada estado. Ressalta-se ainda que a partir de qualquer estado é possível deslocar-se para outro, visto que quem é responsável pelo controle do estado atual é uma *thread* [24], a qual recebe e trata as instruções provenientes da interface web, que será descrita a seguir.

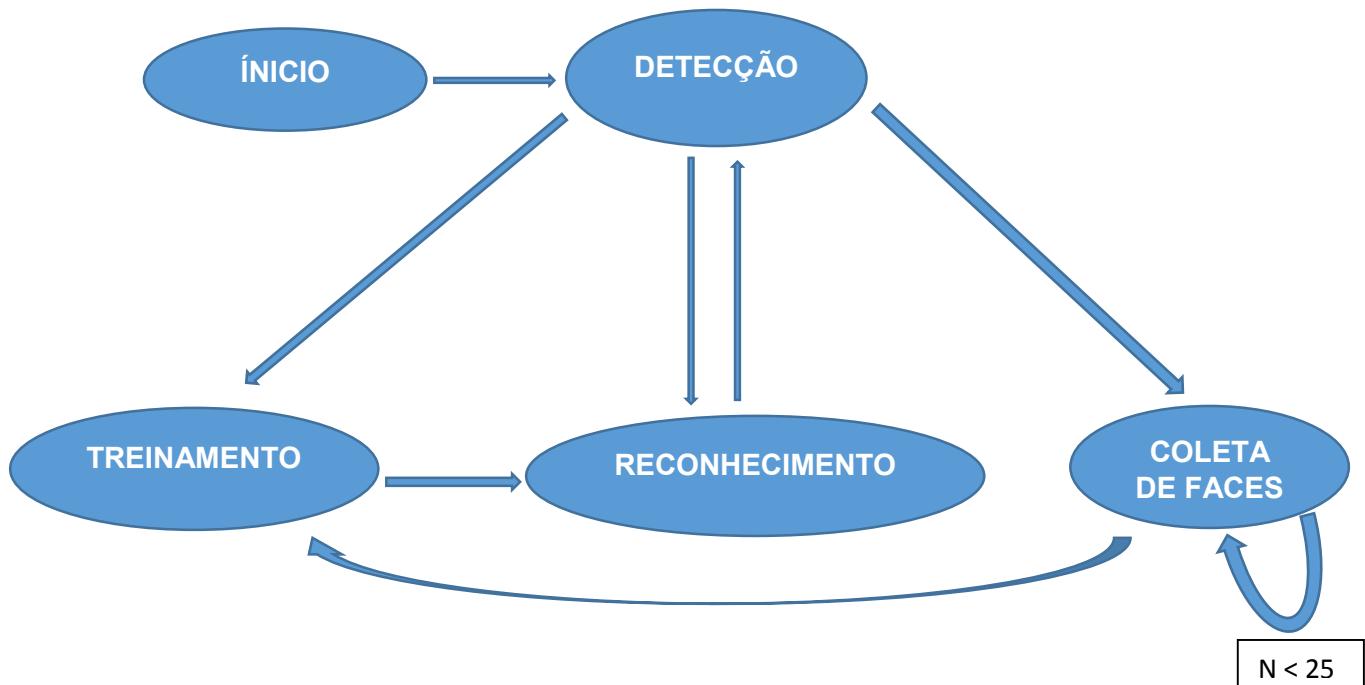


Figura 7 - Diagrama de estados simplificado

A praticidade da biblioteca *OpenCV* foi de suma importância para execução deste projeto, conforme pode ser visto em [25] existem várias rotinas prontas para facilitar o desenvolvimento de aplicações que realizam detecção e reconhecimento de faces. Dentre as quais, destaca-se a possibilidade de utilizar classificadores tipo *Haar* e tipo *LBP* que estão disponíveis em diferentes versões na *internet* em arquivos *XML*.

Para este projeto utilizou-se o arquivo “lbpcascade_frontalface.xml” que contém o classificador *LBP*, como também os arquivos “haarcascade_eye.xml”, “haarcascade_eye_tree_eyeglasses.xml” e “haarcascade_frontalface_alt_tree.xml” que possuem os classificadores tipo *Haar* para detectar olhos, olhos com óculos e faces frontais, respectivamente. Todos arquivos são encontrados na pasta “/home/ubuntu/victor/NewFaceRec”, e foram extraídos do código fonte adquirido online, conforme a seção 4.2.

Após carregar os classificadores acima, a aplicação inicia-se com a leitura das imagens geradas pelo *MJPEG-Streamer*, e inicia-se um pré-processamento. Tal técnica é utilizada para equalizar as imagens antes de iniciar o reconhecimento. Este procedimento tem por intuito remover ruídos utilizando filtros bilaterais, converter a imagem colorida para escala de cinza, padronizar o brilho utilizando equalização por histograma, escalar a imagem para 70x70, detectar a face e aplicar uma máscara elíptica para obter somente o núcleo da face [22].

Ao término da etapa anterior, tem-se a imagem pronta para realizar o reconhecimento. Sendo assim, emprega-se o modelo treinado no modo *TREINAMENTO* e reconstrói-se uma imagem sintética baseando-se no modelo (conjunto de *EigenFaces*) e na imagem de entrada. Por fim, compara-se ambas imagens e calcula-se a similaridade entre ambas.

Nesta etapa, definiu-se um valor de *Threshold* para classificar a imagem como conhecida ou desconhecida. Caso a similaridade calculada seja maior ou igual que o *Threshold* a imagem é classificada como desconhecida (*Unknown*) e retorna *label* “-1”, caso a mesma seja menor que o *Threshold* a imagem é classificada como conhecida e a função retorna a *label* associada a imagem que é utilizada para identificar a qual pessoa pertence esta imagem. Tal *label* é uma marcação inserida na imagem coletada e é passada para a função que realiza o treinamento do banco de dados de imagem no modo *TREINAMENTO*.

Como exemplo, ao inserir a primeira pessoa no banco de imagens, as 50 imagens inseridas terão *label* 0.

Para a construção do banco de imagens a aplicação entra no modo *COLLECT_FACES*. Tal modo irá coletar 25 imagens da pessoa, espelhá-las, processá-las conforme foi descrito anteriormente e salvá-las em um arquivo *XML* o qual será a base de dados. A princípio pensou-se em integrar tal sistema com um servidor de banco de dados, porém diante da facilidade de operar arquivos *XML* com a *API* fornecida pela biblioteca *OpenCV* e também dadas as limitações de recursos encontradas na *beaglebone*, optou-se por não fazer essa implementação neste projeto. Todavia, é sabido das limitações impostas pelo uso de arquivos *XML*, desde ao tamanho crescente do arquivo conforme novos usuários são inseridos no sistema, como também do tempo gasto para leitura e escrita no dispositivo de armazenamento.

4.6. Interface Web de controle

A interface *web* desenvolvida utiliza o servidor *Apache* descrito anteriormente, em conjunto com o *framework php5*. Baseando-se na página que é distribuída no *mjpg-streamer* [21], criou-se um conjunto de páginas *PHP*, conforme a Figura 8. No menu à esquerda há as seguintes seções com as respectivas descrições:

Home: Página inicial que descreve o projeto sucintamente encontrada na Figura 8.



Figura 8 - Página web da interface de controle

Estático: Apresenta um *snapshot* do vídeo proveniente da *webcam*, encontrado na Figura 9.

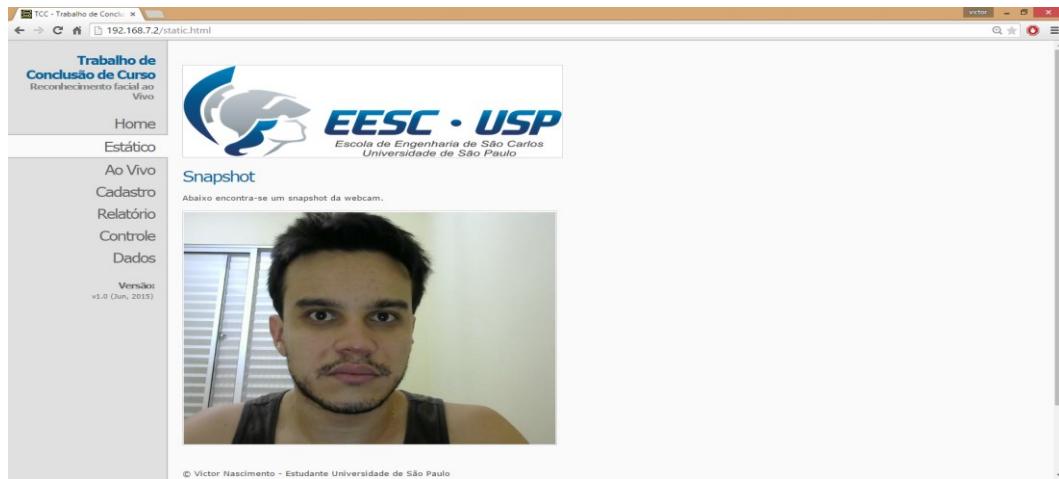


Figura 9 - Página web da aba Estático

Ao Vivo: Disponibiliza o vídeo ao vivo da *webcam* à esquerda como também a última pessoa detectada à direita e o nome da pessoa identificada (se ocorrer), logo em baixo, conforme a Figura 10. Observa-se que na imagem à direita há uma miniatura da imagem reconstruída sinteticamente e uma barra ao lado que representa a similaridade entre ambas imagens. Tal similaridade mede o quanto semelhante a imagem capturada e a imagem gerada sinteticamente são. Logo, quanto mais cheia a barra estiver, mais parecida é a imagem capturada com a pessoa presente no banco de imagens.

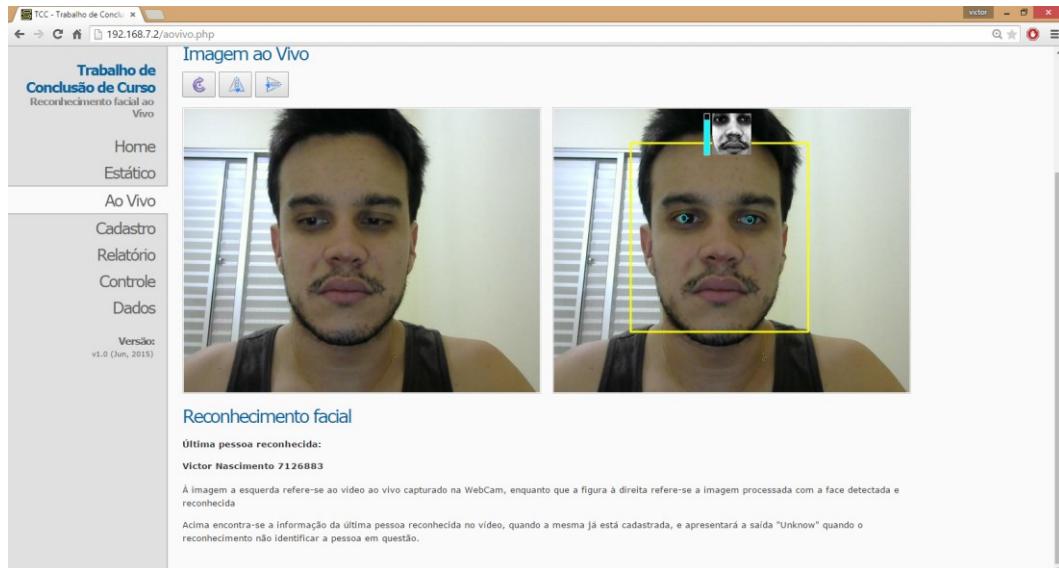


Figura 10 - Página web da aba Ao Vivo.

Cadastro: Esta seção é responsável por gerenciar usuários, sendo possível cadastrar e remover usuários do banco de dados conforme a Figura 11.



Figura 11 - Página web da aba Cadastro

Relatório: Esta seção é responsável por gerenciar os relatórios, permitindo adicionar e remover relatórios do banco de dados, conforme a Figura 12.



Figura 12 - Página web da aba Relatório

Controle: Conforme observa-se na Figura 13, esta seção é responsável por controlar a aplicação em execução na *beaglebone black*. Possibilitando iniciar, fechar e ver o status da aplicação. Além de permitir reiniciar a *BBB*.

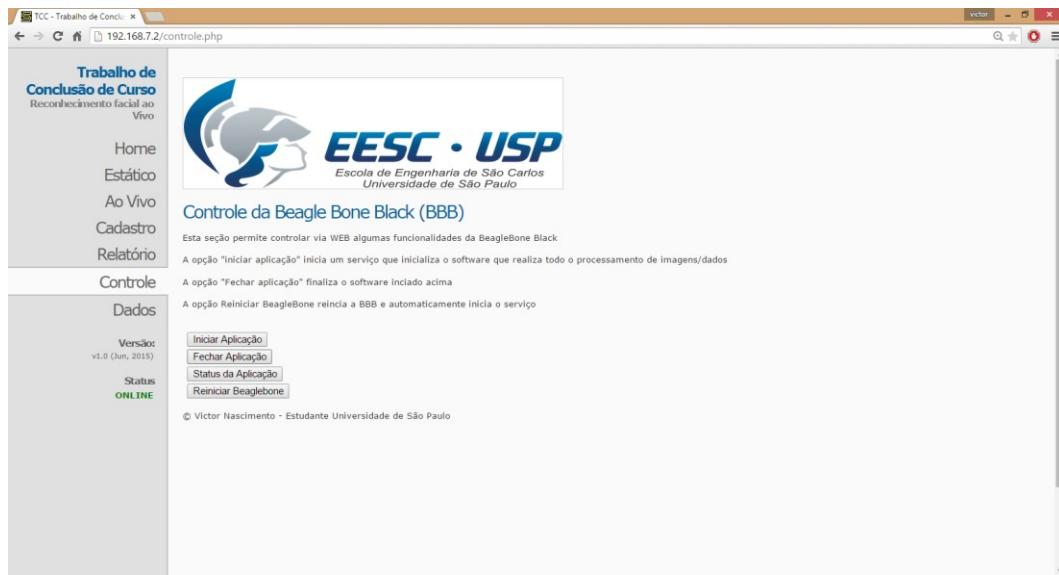


Figura 13 - Página web da aba Controle

Dados: Esta seção disponibiliza os dados presente no banco de dados de usuários e de relatórios, conforme é observado na Figura 14.



Figura 14 - Página web da aba Dados

4.7. Integração da Interface Web de controle e do Software em C++.

Como observado, dois módulos principais que operam de maneira independente compõem este sistema: a interface web de controle e o *software em C++*. Todavia, há a necessidade de estabelecer a comunicação entre ambos, por isso utilizou-se um recurso disponível nos ambientes *Linux*, chamado *FIFO*. Esta estrutura de dados também conhecida como fila, opera de forma que o primeiro elemento inserido será o primeiro a sair. Dessa forma, quando a interface *web* deseja enviar um comando para o *software em C++*, ela escreve a mensagem na *FIFO*. Do outro lado, existe uma *thread* em *loop*, com o nome de “*thread_get_message_web*” encontrada no apêndice B, a qual verifica a cada segundo se existe alguma mensagem nova na *FIFO*. Quando ocorre a leitura de uma mensagem, a mesma é tratada e toma-se uma decisão (alterar o estado da aplicação). A Figura 15 ilustra esta comunicação.

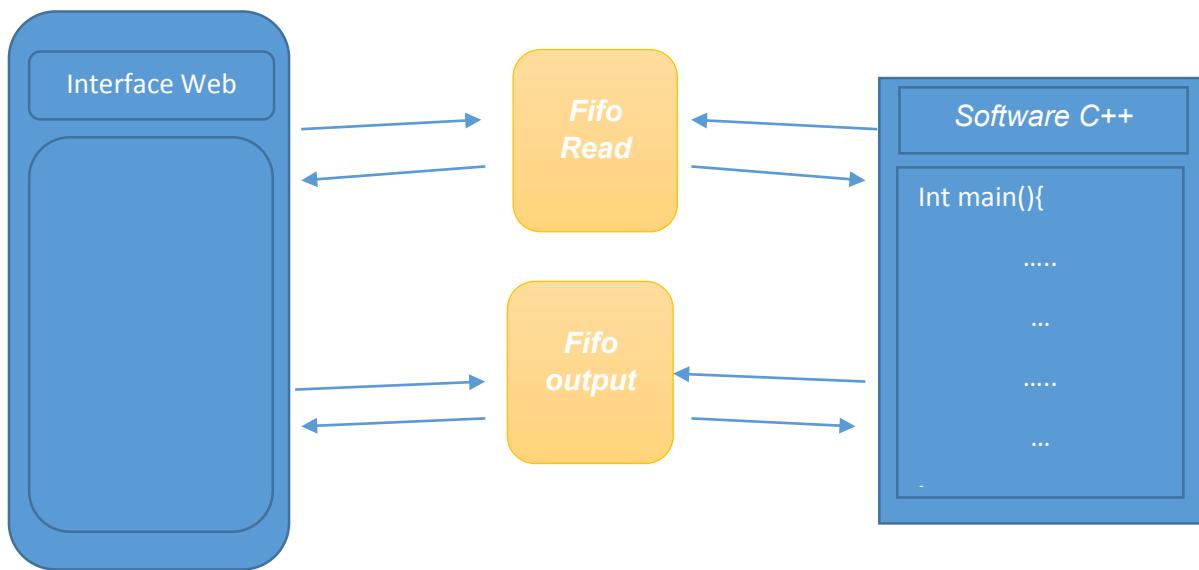


Figura 15 - Diagrama que representa a comunicação entre interface web/Software em C++

Conforme visto no diagrama acima, existem 2 *FIFOs*, cada qual estabelece um canal de comunicação.

FifoRead: Utilizada para enviar informações da interface *web* para o *software em C++* relacionadas ao gerenciamento de usuários e relatórios, bem como do gerenciamento da *BBB* pela aba controle.

FifoOutput: Utilizada para enviar informações do *software em C++* para a interface *web* sobre a pessoa identificada.

O código completo da aplicação, tanto o que se refere ao *software* em C++ como a interface *web* de controle, encontra-se disponível em <https://github.com/vnasceng/newFaceRec>.

5. Resultados

Após a instalação e a configuração das ferramentas, do término do desenvolvimento da interface web e do *software* em C++ e terminada a integração de ambos os módulos, pode-se iniciar a fase de testes e analisar a performance da *beaglebone black* para aplicações com processamento de imagens e de vídeos, bem como analisar a eficiência dos algoritmos de detecção de faces já mencionados. Resultados estes que serão discutidos nas seções subsequentes.

Primeiramente, testar uma aplicação que envolve reconhecimento facial não é uma tarefa trivial, visto que seria necessário depender de voluntários para coletar várias fotos para compor o banco de imagens. Contudo, existem banco de imagens [26] disponíveis na *internet*, os quais podem ser utilizados para realizar os testes. Sendo assim, foi desenvolvida uma versão modificada do *software* que aceite sequências de imagens na entrada, ao invés de utilizar imagens capturadas da *webcam*. Além disso, constrói-se uma tabela composta pelo nome do arquivo de entrada, da pessoa identificada, da similaridade, do tempo necessário para detectar e identificar a pessoa e exibe o sucesso ou falha na identificação. Tais tabelas foram colocadas nos apêndices D e E devido ao seu tamanho, porém foram extraídos gráficos encontrados nas seções subsequentes.

O banco de imagens utilizado é composto por 15 pessoas, das quais os 4 primeiros das tabelas dos apêndices D e E foram de voluntários, isto é, realizou-se a captura de imagens com a *webcam*. As 11 outras pessoas testadas são provenientes de banco de dados online [26]. Para cada pessoa foram utilizadas 35 fotos, todavia, nas tabelas constam apenas as fotos que o algoritmo conseguiu detectar o rosto para então prosseguir com o reconhecimento. Como cada iteração pode detectar números diferentes de fotos, isto é, o total de amostras não é fixo, optou-se por utilizar porcentagem nas análises subsequentes. O total de amostras para cada teste, é encontrado ao final de cada tabela. O número de 35 fotos escolhidos foi definido levando-se em consideração que algumas imagens não seria possível detectar as faces e logo não seria possível realizar a identificação.

5.1. Análise de processamento

Conforme discutido na seção 3.1.1, os recursos disponíveis na *BeagleBone Black* são limitados para esta aplicação, o que pode ser comprovado com os testes abaixo.

Primeiramente, ressalta-se que o sistema operacional Linux não foi alterado, isto é, todas aplicações previamente instaladas foram mantidas e não houve alterações nas prioridades de execução da aplicação. Deste modo, executou-se o software modificado para testes mencionado anteriormente e utilizou-se a ferramenta *asr* [27] a qual consegue coletar dados referentes ao processamento de todos os periféricos da plataforma.

Em suma, coletou-se informações sobre o processamento da *CPU*, da memória e a frequência de operação da *Beaglebone Black*, conforme pode ser observado na tabela encontrada no apêndice C e na Figura 16.

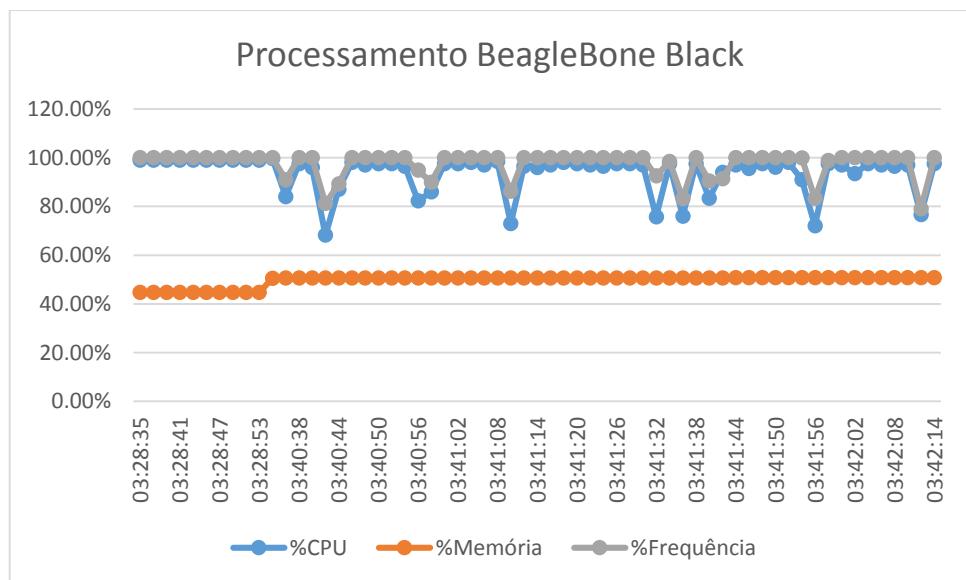


Figura 16 - Gráfico de processamento da *Beaglebone Black*

Observando-se o gráfico da figura 16, infere-se que o processamento da *CPU* está constantemente no seu máximo, assim como a frequência de operação da plataforma (máximo de 1GHz), comportamento este já previsto, pois o processamento de imagens exige muitas operações com matrizes que demandam o uso intensivo do processador. Ao passo que a memória está sendo utilizada na faixa de 50% do total (512MB).

Vale ressaltar que o banco de dados utilizado possui 15 pessoas cadastradas e o uso da memória é proporcional a tal número, logo este sistema terá um limite quanto a

capacidade de usuários de acordo com a memória disponível. Posteriormente, na conclusão, serão propostas alternativas para contornar este problema.

5.2. Análise algoritmo de detecção de Viola Jones

A análise e a comparação dos algoritmos basearam-se em 2 fatores. Primeiro, compara-se o tempo decorrido para processar e identificar a face. Segundo, compara-se a porcentagem de sucesso e falha da identificação de pessoas. Para tanto, utilizou-se 3 valores de *Threshold* (0.5, 0.6 e 0.7) para entender o comportamento da aplicação com a alteração deste limitante. Lembrando-se que quanto maior o valor escolhido, maior a quantidade de imagens serão classificadas como conhecidas.

Extraindo-se os dados da tabela do apêndice D, tem-se a Figura 17 e a Figura 18 que representam a média de tempos (s) e o número de sucesso e de falhas na identificação das pessoas em relação ao valor de *Threshold*, respectivamente, para o algoritmo de detecção que utiliza o classificador *haar cascade*.

Analizando o tempo, identifica-se que o valor de *Threshold* não acarreta grandes mudanças, o que era esperado já que tal valor somente é usado para classificar a imagem como conhecida/desconhecida e não interfere no tempo do processo de detecção e de identificação.

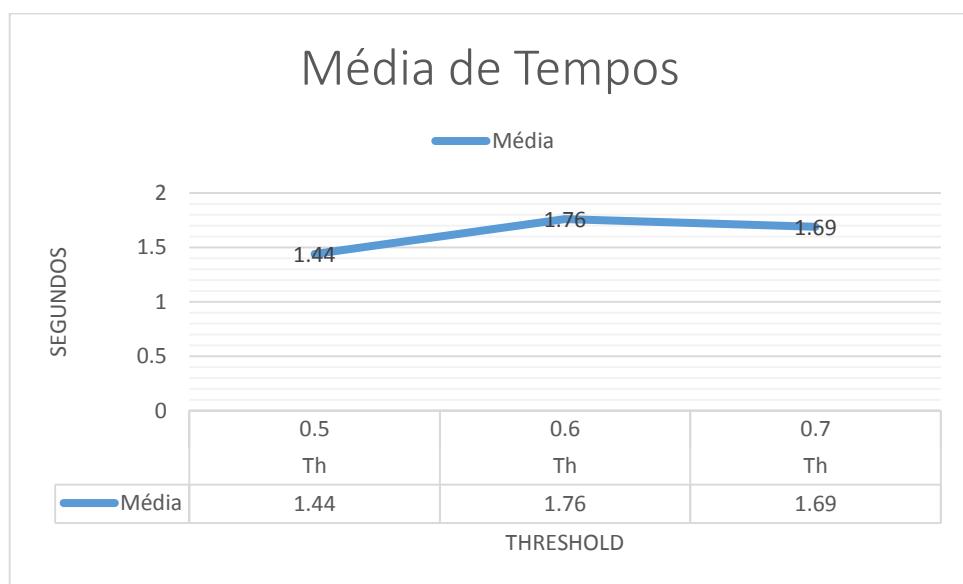


Figura 17 - Gráfico da média de tempos(s) x Threshold do algoritmo que usa Haar Cascade

Observando-se a Figura 18, nota-se claramente que com o aumento de valor do *Threshold* acarreta numa melhora na eficiência do sistema, o número de pessoas identificadas com sucesso aumenta e o número de falhas decresce. De certa forma, isso também já era esperado, visto que o aumento do valor de *Threshold* intensifica o número de imagens classificadas como reconhecidas e neste sistema a maioria das imagens utilizadas para testes também estão presentes no banco de imagens cadastradas, isto é, as imagens testadas fazem parte do banco de imagens, logo as chances de serem identificadas são grandes. Todavia, observando-se a tabela 3 do apêndice D, a qual corresponde ao valor de $Threshold = 0.7$, verifica-se que as únicas imagens de testes de pessoas que não estão cadastradas no sistema (Arquivos de entrada com nome de “JapaFat” e “JapaHair”), foram as que mais falharam nos testes, ou seja, foram identificadas como pessoas conhecidas enquanto que na verdade não deveriam. Isso se deve ao fato do elevado valor de *Threshold*.

Vale ressaltar que o valor de *Threshold* elevado pode ser interessante quando se trata em um cenário onde a maioria dos usuários da aplicação é composto por usuários cadastrados no sistema, como em uma sala de aula onde os alunos que tentarão a identificação já estarão cadastrados. Entretanto, quando trata-se de um sistema de controle como uma catraca, um valor de *Threshold* baixo é mais recomendado, pois para manter a segurança do sistema é preferível que falhe na identificação e não libere o acesso do que liberar o acesso a uma pessoa não autorizada.

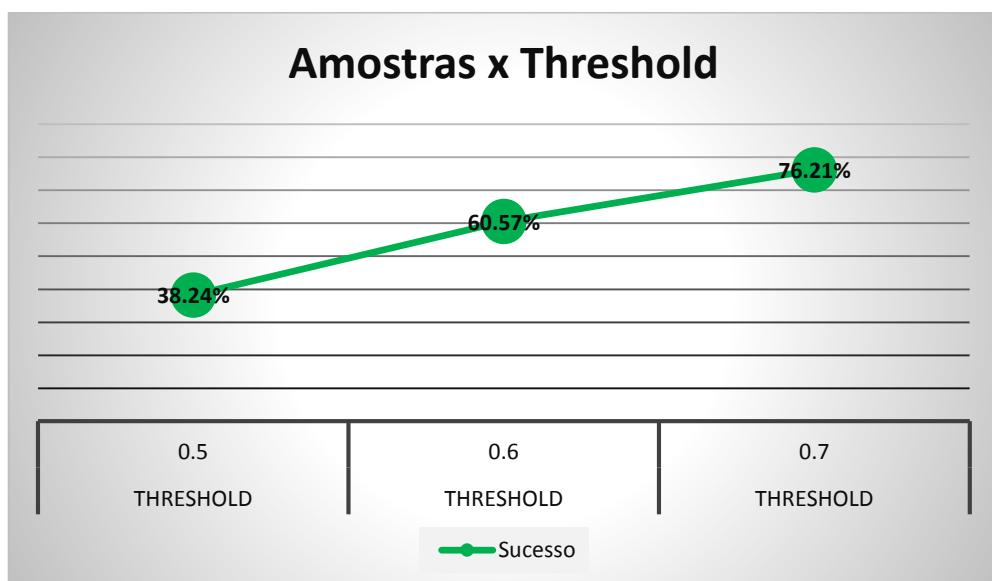


Figura 18- Gráfico da porcentagem de sucesso/falha na identificação em função do *Threshold* do algoritmo que usa *Haar Cascade*.

5.3. Análise do algoritmo de detecção com *LBP*

Análogo à seção 5.2, realizou-se a execução dos mesmos testes, porém agora com o algoritmo *LBP* ao invés do classificador *Haar* do algoritmo de *Viola Jones*, obteve-se as tabelas do apêndice E, as quais foram utilizadas para extrair os gráficos da figura 19 e 20.

Na Figura 19 tem-se a média dos tempos em relação ao *Threshold*. É semelhante ao que foi encontrado anteriormente o valor de *threshold* não acarreta grandes alterações no tempo gasto para identificação da face, já que o intervalo encontrado foi de 0,12 segundos entre mínimo e o máximo valor de *Threshold*. Entretanto, nota-se que há uma melhora no desempenho quando compara-se *LBP* x *Haar*, já que o primeiro exige na faixa de 1,2 segundos enquanto que o segundo exige na faixa de 1,5 segundos para realizar a detecção e identificação de uma face.

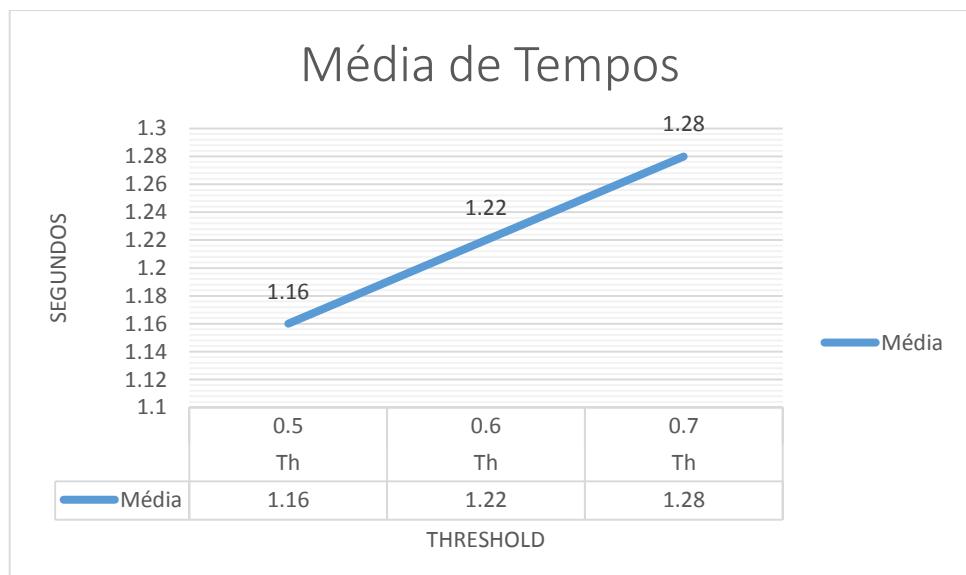


Figura 19 - Gráfico da média de tempos(s) x Threshold do algoritmo de detecção que usa *LBP*

Similarmente ao que também foi evidenciado no algoritmo que utiliza classificadores tipo *Haar Cascade*, a identificação de faces possui melhor eficiência com o aumento do *Threshold* quase que na mesma proporção, conforme pode ser observado na Figura 20. Além disso, nota-se que a taxa de sucesso para os três valores de *Threshold* utilizando-se o *LBP* é maior que as taxas encontradas na avaliação anterior.

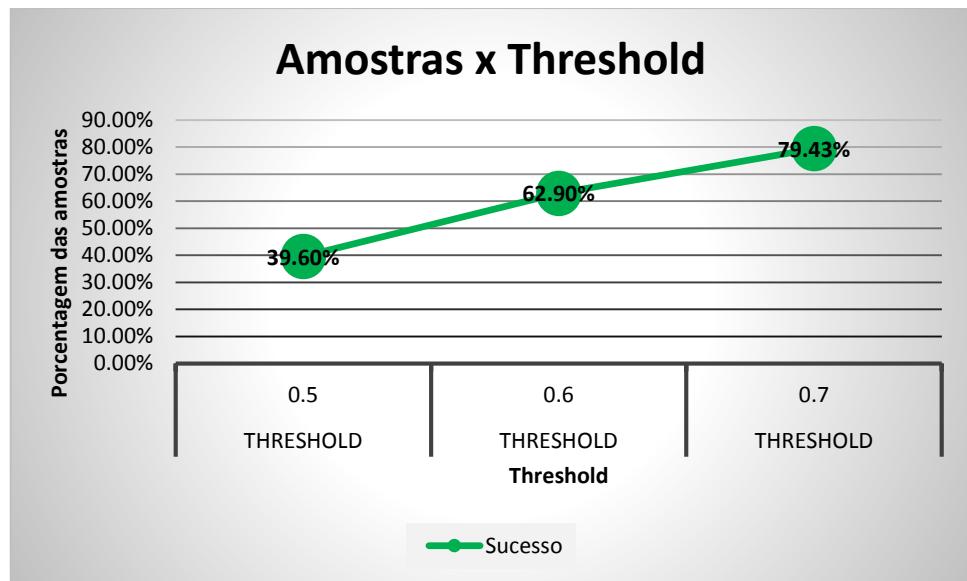


Figura 20- Gráfico da porcentagem de sucesso/falha na identificação em função do *Threshold* do algoritmo que usa *LBP*

5.4. Relatório de presença

O relatório de presença é gerado automaticamente quando existe uma tarefa cadastrada. O intuito deste é simular uma lista de presença, logo contém o nome da disciplina e número da turma, o professor responsável, a hora inicial e hora final da disciplina bem como os alunos identificados neste período. Conforme observa-se na Figura 21 tal relatório ilustra com clareza a presença dos alunos e seria uma alternativa para as atuais listas de presenças preenchidas à mão.

A	B
1 Relatório gerado:	21:12
2 Disciplina:	SEL630
3 Turma: 2	
4 Professor Responsável:	Evandro Rodrigues
5 Início: 21:11	Fim: 21:12
6	
7 Aluno	Nro. USP
8 VictorNascimento	7126883
9 Robson Silva	11111114
10 Lucas Telles	11111111
11 Total:	3

Figura 21 - Relatório de presença gerado automaticamente

6. Conclusões

Ao término deste projeto e diante dos resultados obtidos no capítulo 5, pode-se inferir que os objetivos iniciais propostos para o projeto foram alcançados. Utilizando-se *Linux* embarcado, desenvolveu-se uma solução de baixo custo e baixa complexidade que realiza a detecção e reconhecimento facial com o auxílio da biblioteca *OpenCV*. Além disso, empregou-se técnicas de tratamento de multimídia, estudou-se e aplicou-se conceitos e algoritmos de visão computacional e fez-se uma análise desta solução em uma plataforma embarcada.

Observando os resultados adquiridos, considerando-se que tanto o tempo médio de processamento para identificação como a taxa de sucesso na identificação apresentaram melhores resultados com a utilização da detecção de imagens por meio do algoritmo de *LBP*, pode-se concluir que a aplicação embarcada na *Beaglebone Black* a qual realiza a identificação facial apresenta melhor desempenho quando comparado com o algoritmo de *Viola Jones*.

Além disso, pode-se concluir que o tempo exigido para identificação de uma face não é elevado, quando comparado com outros métodos de identificação tradicionais como apresentar um documento de identidade ou mesmo utilizar um cartão eletrônico em uma leitora, já que esta solução de identificação facial demora em média 1,2 ou 1,5 segundos.

Nota-se ainda que a utilização de uma interface *Web* de controle agregou um baixo custo ao sistema. Primeiro porque tal solução dispensa o uso de um dispositivo de interface gráfica como um display *LCD*. Segundo, pois reduziu a complexidade do desenvolvimento, visto que o desenvolvimento de uma *GUI* em sistemas embarcados não é trivial.

Uma sugestão para trabalhos futuros é tentar integrar um servidor de banco de dados nesta solução. Para tanto, poderia utilizar uma outra solução de *hardware*, com mais recursos, para realizar este processamento. Esta solução não necessariamente deveria ser um sistema embarcado, poderia ser um servidor *Linux* em um *desktop*. Desta forma solucionaria os gargalos de processamento, memória e espaço em disco. Ficando a cargo da *Beaglebone* apenas capturar e processar as imagens para serem repassadas via algum protocolo de comunicação, como *UDP*, para o servidor *Linux*. Este último apenas devolveria a resposta final, indicando o reconhecimento ou não. Tal servidor poderia inclusive responder à várias requisições de várias *BeagleBones*, cenário este encontrado em um bloco didático com várias salas de aula e cada sala com sua plataforma de reconhecimento.

Espera-se ainda que com o avanço contínuo dos dispositivos eletrônicos, o processamento de multimídia se tornará simples e extremamente rápido, possibilitando que soluções embarcadas tornem-se cada vez mais comuns e viáveis, facilitando as tarefas do dia-a-dia.

Referências bibliográficas

- [1] *P.I. Okwu and I.N. Onyeje* "Ubiquitous Embedded Systems Revolution: Applications and Emerging Trends", *International Journal of Engineering Research and Applications (IJERA)* Vol. 3, Issue 4, Jul-Aug 2013, pp.610-616
- [2] *Beagleboard.org*, 'BeagleBoard.org - black', 2015. [Online]. Disponível: <http://beagleboard.org/black>. [Acesso em 28 de maio de 2015].
- [3] *Opencv.org*, 'OpenCV | OpenCV', 2015. [Online]. Disponível: <http://opencv.org>. [Acessado em 28 de maio de 2015].
- [4] *M. Mitchell, J. Oldham and A. Samuel*, *Advanced Linux programming*. Indianapolis, Ind.: New Riders Pub., 2001.
- [5] R. Silva and E. Alecrim, 'A tecnologia por trás da Rover Curiosity | Tecnoblog', *Tecnoblog*, 2012. [Online]. Disponível: <https://tecnoblog.net/109317/tecnologia-rover-curiosity/>. [Acesso em 01 de maio de 2015].
- [6] E. Barros e S. Cavalcanti Introdução aos Sistemas Embarcados. Disponível: <http://www.cin.ufpe.br/~vba/periodos/8th/s.e/aulas/STP%20%20Intro%20Sist%20Embarcados.pdf> [Acesso em 01 de maio de 2015].
- [7] *Intel.com*, 'Intel® Embedded Microcontrollers', 2015. [Online]. Disponível: <http://www.intel.com/design/embcontrol/>. [Acesso em 20 Maio 2015.]
- [8] *Microchip.com*, 'Microchip MCUs | Microchip Technology Inc.', 2015. [Online]. Disponível: <http://www.microchip.com/pic>. [Acessado em 28 de maio de 2015].
- [9] *Arm.com*, 'ARM Processor Architecture - ARM', 2015. [Online]. Disponível: <http://www.arm.com/products/processors/instruction-set-architectures/index.php>. [Acesso em 20 Maio 2015].
- [10] *M. Mitchell, J. Oldham and A. Samuel*, *Advanced Linux programming*. Indianapolis, Ind.: New Riders Pub., 2001.
- [11] *Gnu.org*, 'The GNU Operating System and the Free Software Movement', 2015. [Online]. Disponível: <http://www.gnu.org/>. [Acesso em 01 de maio de 2015].
- [12] *Ming-Hsuan Yang, D. Kriegman and N. Ahuja*, 'Detecting faces in images: a survey', *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 1, pp. 34-58, 2002.

- [13] S. DEVI S; P. K Mane and AjayKumar D "Face Detection System using OpenCV on Beagle Board"
- [14] P. Viola and M. Jones, 'Rapid object detection using a boosted cascade of simple features', *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001.
- [15] G. Araújo, 'Algoritmo para reconhecimento de características faciais baseado em filtros de correlação', Mestrado, UFRJ/COPPE, 2010.
- [16] Y. Rodrigues, 'Face Detection and Verification using Local Binary Patterns', *Phd, École Polytechnique Fédérale de Lausanne*, 2006.
- [17] T. Ojala, M. Pietikainen and T. Maenpaa, 'Multiresolution gray-scale and rotation invariant texture classification with local binary patterns', *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 7, pp. 971-987, 2002.
- [18] M. Turk and A. Pentland, 'Eigenfaces for Recognition', *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
- [19] Angstrom-distribution.org., 'The Ångström Distribution | supporting everything from toasters to servers', 2015. [Online]. Disponível: <http://angstrom-distribution.org>. [Acessado em 28 de maio de 2015].
- [20] Apache.org, 'Welcome to The Apache Software Foundation!', 2015. [Online]. Available: <http://apache.org/>. [Acessado em 28 de maio de 2015].
- [21] GitHub, 'jacksonliam/mjpg-streamer', 2015. [Online]. Disponível: <https://github.com/jacksonliam/mjpg-streamer>. [Acessado em 28 de maio de 2015].
- [22] Shervinemami.info, 'Face Detection and Face Recognition with Real-time Training from a Camera', 2015. [Online]. Disponível: <http://www.shervinemami.info/faceRecognition.html>. [Acessado em 28 de maio de 2015].
- [23] S. Emami, *OpenCV 2 Hotshot*. Birmingham: Packt Publishing, Limited, 2012.
- [24] Computing.llnl.gov, 'POSIX Threads Programming', 2015. [Online]. Disponível: <https://computing.llnl.gov/tutorials/pthreads/>. [Acessado em 28 de maio de 2015].
- [25] Docs.opencv.org, 'Face Recognition with OpenCV — OpenCV 2.4.11.0 documentation', 2015. [Online]. Disponível: http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html. [Acessado em 28 de maio de 2015].

[26] 'Banco de dados de Imagens', 2015. [Online]. Disponível: http://robotics.csie.ncku.edu.tw/Databases/ FaceDetect_PoseEstimate.htm#Our_Database_. [Acessado em 28 de maio de 2015].

[27] *Linux.die.net*, 'sar(1) - Linux man page', 2015. [Online]. Disponível: <http://linux.die.net/man/1/sar>. [Acessado em 28 de maio de 2015].

APÊNDICE A - Instalação e configuração do Ubuntu 14 LTS na BBB

1. Instalação

A instalação de uma imagem na *Beagle* ocorre por meio da escrita da imagem no cartão de memória *microSD* e em seguida da realização do procedimento de *boot* descrito na seção 1.2.

Vale ressaltar que há dois tipos de imagens do *Ubuntu*. A primeira, é compilada para ser escrita e executada na *flash* (*eMMC*). A segunda, é compilada para realizar o *boot* pelo cartão de memória. Sendo assim, a primeira apresenta um tamanho reduzido se comparada com a segunda. Neste projeto optou-se por utilizar o *Ubuntu* a partir da *flash*, para que não fosse necessário pressionar o botão *boot* para realizar o *boot* (como é exigido com *boot* a partir do cartão *microSD*). Tal característica pode ser um problema, já que em uma eventual queda de energia seria necessário um usuário pressionar o botão de *boot* para que o sistema voltasse a operar.

Primeiramente, realiza-se o download da imagem pré-compilada e então a escreve no cartão de memória.

Para o download utiliza-se:

```
wget https://rcn-ee.net/deb/flasher/trusty/BBB-eMMC-flasher-ubuntu-14.04-console-armhf-2014-08-13-2gb.img.xz
```

E para a escrita, utiliza-se:

```
sudo dd if=./BBB-eMMC-flasher-ubuntu-14.04-console-armhf-2014-08-13-2gb.img of=/dev/sdX
```

Onde “X” deve ser trocado pela letra que o cartão de memória recebeu do sistema operacional. Para identificação, pode-se utilizar o comando “fdisk –l” que lista todos os dispositivos de armazenamento.

Nesse ponto pode-se realizar o procedimento de *boot* descrito no início desta seção 1.2.

1.1. Procedimento de *Boot*

O *Boot* da *Beagle* pode ser realizado de duas maneiras. Caso queira-se realizar o *boot* via o cartão *microSD* o procedimento é encontrado na seção 1.1.1, e caso queira realiza-lo via *eMMC*, o procedimento é encontrado na seção 1.1.2.

1.1.1. *Boot MicroSD*

- Inserir o cartão *microSD* na *Beagle*.
- Pressionar o botão *Boot*.
- Inserir a alimentação de energia.
- Espera-se os 4 *LEDs* se acenderem e começarem a piscar e soltar o botão de *Boot*.
- Aguardar a tela de *boot*

1.1.2. *Boot eMMC*

- Inserir o cartão *microSD* na *Beagle*.
- Pressionar o botão *Boot*.
- Inserir a alimentação de energia.
- Espera-se os 4 *LEDs* se acenderem e soltar o botão de *Boot*.

- Espera-se até que todos os 4 *LEDs* estejam acessos permanentemente.
- Retira-se a alimentação de energia e retira-se o cartão
- O próximo *boot* ocorrerá pela *eMMC*.

Já na tela de boot, esta imagem do *Ubuntu* tem por padrão:

Login: **ubuntu**

Senha: **temppwd**

A troca de *password* pode ser feita via linha de comando, utilizando “*sudo passwd*”.

2. Comunicação com a *Beagle*

A comunicação com a *Beagle* pode ser realizada de diversas maneiras dentre as quais tem-se: *SSH*, serial ou mesmo conectando-a em um monitor de TV. Por simplicidade, optou-se por utilizar a conexão remota via *SSH* para conectar-se a *Beagle*. Dessa forma, ao conectar a *Beagle* ao *desktop* via *USB* cria-se uma conexão local interna entre a *Beagle* e o computador, sendo que a primeira tem *IP* 192.168.7.2 e o segundo 192.168.7.1. Utilizando-se de um programa de *SSH* é possível conectar-se a *Beagle* utilizando *login*: *ubuntu@192.168.7.2* e senha: *temppwd*.

A fim de testar se a comunicação está correta, pode-se utilizar o comando “*ping 192.168.7.2*” ou mesmo acessar este *IP* via *browser* o qual deve mostrar uma página *WEB* padrão da *BeagleBone Black*, conforme a figura abaixo

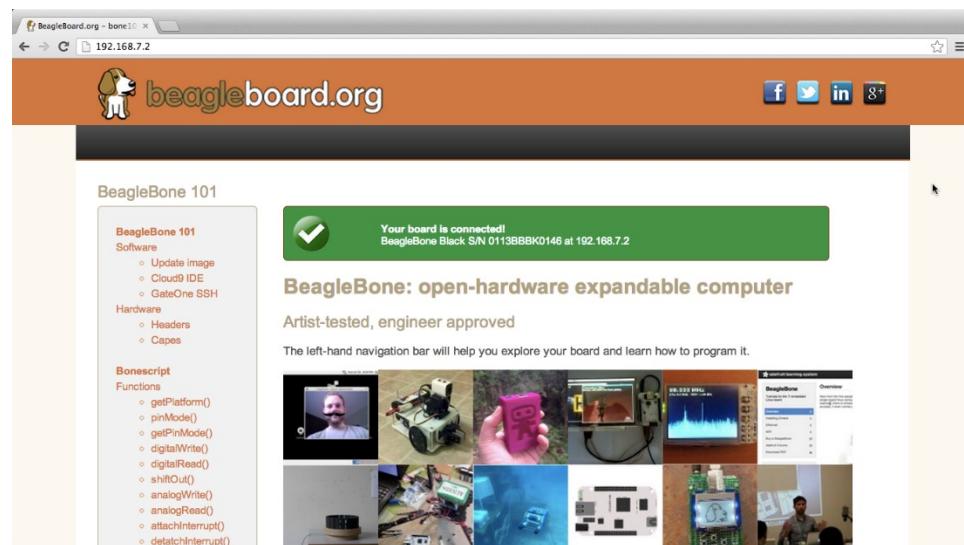


Figura 4.3.1 – Imagem inicial BeagleBone Black

APÊNDICE B - Código Software em C++

```

/*
*****
* Face Recognition using Eigenfaces or
Fisherfaces
*****
* by Shervin Emami, 5th Dec 2012
* http://www.shervinemami.info/openCV.html
*****
* Ch8 of the book "Mastering OpenCV with
Practical Computer Vision Projects"
* Copyright Packt Publishing 2012.
* http://www.packtpub.com/cool-projects-with-
opencv/book
*****
****

///////////////////////////////
// WebcamFaceRec.cpp, by Shervin Emami
//(www.shervinemami.info) on 30th May 2012.
// Face Detection & Face Recognition from a webcam
using LBP and Eigenfaces or Fisherfaces.
///////////////////////////////
// Some parts are based on the tutorial & code by
Robin Hewitt (2007) at:
// "http://www.cognitics.com/opencv/servo_2007_series/
part_5/index.html"
// Some parts are based on the tutorial & code by
Shervin Emami (2009) at:
// "http://www.shervinemami.info/faceRecognition.html"
// Requires OpenCV v2.4.1 or later (from June
2012), otherwise the FaceRecognizer will not
compile or run.
// ///////////////////////////////
// The Face Recognition algorithm can be one of
these and perhaps more, depending on your version
of OpenCV, which must be atleast v2.4.1:
// "FaceRecognizer.Eigenfaces": Eigenfaces,
also referred to as PCA (Turk and Pentland, 1991).
// "FaceRecognizer.Fisherfaces": Fisherfaces,
also referred to as LDA (Belhumeur et al, 1997).
// "FaceRecognizer.LBPH": Local Binary
Pattern Histograms (Ahonen et al, 2006).
//char *facerecAlgorithm =
"FaceRecognizer.Fisherfaces";
char *facerecAlgorithm =
"FaceRecognizer.Eigenfaces";
// char *facerecAlgorithm = "FaceRecognizer.LBPH";

// Sets how confident the Face Verification
algorithm should be to decide if it is an unknown
person or a known person.
// A value roughly around 0.5 seems OK for
Eigenfaces or 0.7 for Fisherfaces, but you may want
to adjust it for your
// conditions, and if you use a different Face
Recognition algorithm.
// Note that a higher threshold value means
accepting more faces as known people,
// whereas lower values mean more faces will be
classified as "unknown".
const float UNKNOWN_PERSON_THRESHOLD = 0.7f;

// Cascade Classifier file, used for Face
Detection.
const char *faceCascadeFilename =
"lbpcascade_frontalface.xml"; // LBP face
detector.

//const char *faceCascadeFilename = " "; // Haar
face detector.
//const char *eyeCascadeFilename1 =
"haarcascade_lefteye_2splits.xml"; // Best eye
detector for open-or-closed eyes.
//const char *eyeCascadeFilename2 =
"haarcascade_righteye_2splits.xml"; // Best eye
detector for open-or-closed eyes.
//const char *eyeCascadeFilename1 =
"haarcascade_mcs_lefteye.xml"; // Good eye
detector for open-or-closed eyes.
//const char *eyeCascadeFilename2 =
"haarcascade_mcs_righteye.xml"; // Good eye
detector for open-or-closed eyes.
const char *eyeCascadeFilename1 =
"haarcascade_eye.xml"; // Basic eye
detector for open eyes only.
const char *eyeCascadeFilename2 =
"haarcascade_eye_tree_eyeglasses.xml"; // Basic eye
detector for open eyes if they might wear glasses.

// Set the desired face dimensions. Note that
"getPreprocessedFace()" will return a square face.
const int faceWidth = 70;
const int faceHeight = faceWidth;

// Try to set the camera resolution. Note that this
only works for some cameras on
// some computers and only for some drivers, so
don't rely on it to work!
const int DESIRED_CAMERA_WIDTH = 640;
const int DESIRED_CAMERA_HEIGHT = 480;

// Parameters controlling how often to keep new
faces when collecting them. Otherwise, the training
set could look to similar to each other!
const double CHANGE_IN_IMAGE_FOR_COLLECTION = 0.3;
// How much the facial image should change before
collecting a new face photo for training.
const double CHANGE_IN_SECONDS_FOR_COLLECTION =
1.0; // How much time must pass before
collecting a new face photo for training.

const char *windowName = "WebcamFaceRec"; // Name
shown in the GUI window.
const int BORDER = 8; // Border between GUI
elements to the edge of the image.
int countPickImage = 0;
const bool preprocessLeftAndRightSeparately = true;
// Preprocess left & right sides of the face
separately, in case there is stronger light on one
side.

// Set to true if you want to see many windows
created, showing various debug info. Set to 0
otherwise.
bool m_debug = false;

bool imgfile = false;
bool img = false;

const int NUMBER_COLLECT = 25;
const int NUMBER_FACES_COLLECT = NUMBER_COLLECT*2;

#include <stdio.h>
#include <vector>
#include <string.h>
#include <iostream>
#include <pthread.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <algorithm>
#include <iostream>
#include <string>

#include <iterator>
// Include OpenCV's C++ Interface
#include "opencv2/opencv.hpp"

// Include the rest of our code!

```

```

#include "detectObject.h"           // Easily detect
faces or eyes (using LBP or Haar Cascades).
#include "preprocessFace.h"        // Easily
preprocess face images, for face recognition.
#include "recognition.h"          // Train the face
recognition system and recognize a person from an
image.

#include "ImageUtils.h"            // Shervin's handy
OpenCV utility functions.

using namespace cv;
using namespace std;

#if !defined VK_ESCAPE
#define VK_ESCAPE 0x1B           // Escape character
(27)
#endif

#define VK_0 0x30
#define VK_1 0x31
#define VK_2 0x32
#define VK_3 0x33
#define VK_4 0x34
#define VK_5 0x35
#define VK_6 0x36

pthread_mutex_t mutex1 = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t mutex_web =
PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t mutex_fifoOutput =
PTHREAD_MUTEX_INITIALIZER;

// Running mode for the Webcam-based interactive
GUI program.
enum MODES {MODE_STARTUP=0, MODE_DETECTION,
MODE_COLLECT_FACES, MODE_TRAINING,
MODE_RECOGNITION, MODE_DELETE_ALL,
MODE_IDLE, MODE_EXIT};
const char* MODE_NAMES[] = {"Startup", "Detection",
"Collect Faces", "Training", "Recognition", "Delete
All", "ERROR!"};
MODES m_mode = MODE_STARTUP;

int m_selectedPerson = -1;
int m_numPersons = 0;
int nPersons;
vector<int> m_latestFaces;
vector<Mat> preprocessedFaces;
vector<int> faceLabels;
// array of person names (indexed by the person
number).
vector<string> personNames;
vector<DataStudent> personIdentified;
vector<DataStudent> studentsDB;
string lastPersonName = "";

//report variables
char current_hour[6];
FILE *date_fd;
int numRelatorios = 0;
bool reporting = false;
char fileImage[64];
VideoCapture videoCapture;
int id1 = 2;

// Position of GUI buttons:
Rect m_rcBtnAdd;
Rect m_rcBtnDel;
Rect m_rcBtnDebug;
int m_gui_faces_left = -1;
int m_gui_faces_top = -1;

char collect[3][64]; //variable to save name(0),
lastname(1),numUSP(2)

bool operator==(const DataStudent& std1, const
DataStudent& std2) {
    return (std1.numUSP.compare(std2.numUSP) ==
0);
}

bool operator==(const DataReport& rpt1, const
DataReport& rpt2) {

```

```

    return (
(rpt1.disciplina.compare(rpt2.disciplina) == 0) &&
(rpt1.turma.compare(rpt2.turma) == 0));
}

vector<DataReport> report;

// relatorioData relatorio[10];
char relatorio[5][64]; //variable to save
disciplina(0) turma (1) professor(2) startHour(3)
endHour (4)

// C++ conversion functions between integers (or
floats) to std::string.
template <typename T> string toString(T t)
{
    ostringstream out;
    out << t;
    return out.str();
}

template <typename T> T fromString(string t)
{
    T out;
    istringstream in(t);
    in >> out;
    return out;
}

// private class OnlineData{
//     private string disciplina = new string(64);
//     private string turma = new string(2);
//     private string professor = new string(64);
//     private string startHour = new string(5);
//     private string endHour = new string(5);

//     public OnlineData(string disciplina, string
turma, string professor, string startHour, string
endHour ){
//         this.disciplina = disciplina;
//         this.turma = turma;
//         this.professor = professor;
//         this.startHour = startHour;
//         this.endHour = endHour;
//     }
// }

int checkReport(DataReport rpt){

    for(unsigned i = 0; i < report.size(); i++){
        if(report[i] == rpt){
            cout << "they are equal" << endl;
            return i; //return position
        }else
            cout << "not equal" << endl;
    }
    return -1; //it's not present
}

int removeReport(string disciplina, string turma){
    //create a report to search
    DataReport rpt = DataReport(disciplina,
turma,"","","");
    int ret = checkReport(rpt);

    if(ret >= 0){
        report.erase(report.begin() + ret);
        numRelatorios--;
        storeClassData();
        return 1;
    }else{
        cout << "this report is not present in DB"
<< endl;
    }
    return -1;
}

int checkStudent(DataStudent std){

    int i;
    for(i = 0; i < studentsDB.size(); i++){

```

```

        // cout << "std1: " << studentsDB[i].numUSP
        << " size " << studentsDB[i].numUSP.length() << "
std2 " << std.numUSP << " size " <<
std.numUSP.length() << endl;
        if(studentsDB[i] == std){
            cout << "they are equal" << endl;
            return i; //it's present
        }else
            cout << "not equal" << endl;

    }
    return -1; //it's not present
}

int removeStudent(string numUSP){

    int ret;
    //create an fake student with the number USP
    DataStudent std = DataStudent("", "", numUSP);

    //check if the student is present, if there is
    //a student with this number usp, will return its
    //position
    ret = checkStudent(std);

    cout << "Return " << ret << endl;
    if(ret >= 0){ //this student is present
        studentsDB.erase(studentsDB.begin() + ret);
        //removes this position in students DB

        preprocessedFaces.erase(preprocessedFaces.begin() +
ret*NUMBER_FACES_COLLECT, preprocessedFaces.begin() +
+ ret*NUMBER_FACES_COLLECT + NUMBER_FACES_COLLECT);
        faceLabels.erase(faceLabels.begin() +
ret*NUMBER_FACES_COLLECT, faceLabels.begin() +
ret*NUMBER_FACES_COLLECT + NUMBER_FACES_COLLECT);
        //decrease the labels of each element
        //subsequent of the removed element
        vector<int>::size_type sz =
faceLabels.size();
        for(unsigned i = ret*NUMBER_FACES_COLLECT;
i < sz; i++){
            faceLabels[i] = faceLabels[i] - 1;
        }

        m_latestFaces.erase(m_latestFaces.begin() + ret);
        m_numPersons--;
        storeTrainingData(); //update xml file
        return 1;
    }else{
        cout << "this student is not present in DB"
<< endl;
    }
    return -1;
}

// Load the face and 1 or 2 eye detection XML
// classifiers.
void initDetectors(CascadeClassifier &faceCascade,
CascadeClassifier &eyeCascade1, CascadeClassifier
&eyeCascade2)
{
    // Load the Face Detection cascade classifier
    // xml file.
    try { // Surround the OpenCV call by a
    try/catch block so we can give a useful error
    message!
        faceCascade.load(faceCascadeFilename);
    } catch (cv::Exception &e) {}
    if ( faceCascade.empty() ) {
        cerr << "ERROR: Could not load Face
        Detection cascade classifier [" <<
faceCascadeFilename << "]!" << endl;
        cerr << "Copy the file from your OpenCV
        data folder (eg: 'C:\\OpenCV\\data\\lbpcascades')
        into this WebcamFaceRec folder." << endl;
        exit(1);
    }
    cout << "Loaded the Face Detection cascade
    classifier [" << faceCascadeFilename << "]." <<
    endl;

    // Load the Eye Detection cascade classifier
    // xml file.
}
try { // Surround the OpenCV call by a
try/catch block so we can give a useful error
message!
    eyeCascade1.load(eyeCascadeFilename1);
} catch (cv::Exception &e) {}
if ( eyeCascade1.empty() ) {
    cerr << "ERROR: Could not load 1st Eye
    Detection cascade classifier [" <<
eyeCascadeFilename1 << "]!" << endl;
    cerr << "Copy the file from your OpenCV
    data folder (eg: 'C:\\OpenCV\\data\\haarcascades')
    into this WebcamFaceRec folder." << endl;
    exit(1);
}
cout << "Loaded the 1st Eye Detection cascade
    classifier [" << eyeCascadeFilename1 << "]." <<
    endl;

    // Load the Eye Detection cascade classifier
    // xml file.
    try { // Surround the OpenCV call by a
    try/catch block so we can give a useful error
    message!
        eyeCascade2.load(eyeCascadeFilename2);
    } catch (cv::Exception &e) {}
    if ( eyeCascade2.empty() ) {
        cerr << "Could not load 2nd Eye Detection
        cascade classifier [" << eyeCascadeFilename2 <<
        "]!" << endl;
        // Dont exit if the 2nd eye detector did
        // not load, because we have the 1st eye detector at
        // least.
        //exit(1);
    }
    else
        cout << "Loaded the 2nd Eye Detection
        cascade classifier [" << eyeCascadeFilename2 <<
        "]." << endl;
}

// Get access to the webcam.
void initWebcam()
{
    // Get access to the default camera.
    try { // Surround the OpenCV call by a
    try/catch block so we can give a useful error
    message!
        videoCapture.open(0);
    } catch (cv::Exception &e) {}
    if ( !videoCapture.isOpened() ) {
        cerr << "ERROR: Could not access the
        camera!" << endl;
        exit(1);
    }
    // videoCapture.set(CV_CAP_PROP_FRAME_WIDTH,
    DESIRED_CAMERA_WIDTH);
    // videoCapture.set(CV_CAP_PROP_FRAME_HEIGHT,
    DESIRED_CAMERA_HEIGHT);
}

int closeWebCam(){

    try { // Surround the OpenCV call by a
    try/catch block so we can give a useful error
    message!
        videoCapture.release();
    } catch (cv::Exception &e) {}
    if ( !videoCapture.isOpened() ) {
        cerr << "ERROR: Could not access the
        camera!" << endl;
    }
    return 1;
}

// Draw text into an image. Defaults to top-left-
// justified text, but you can give negative x coords
// for right-justified text,
// and/or negative y coords for bottom-justified
// text.
// Returns the bounding rect around the drawn text.
Rect drawString(Mat img, string text, Point coord,
Scalar color, float fontScale = 0.6f, int thickness
= 1, int fontFace = FONT_HERSHEY_COMPLEX)
{
    // Get the text size & baseline.
    int baseline=0;
}

```

```

    Size textSize = getTextSize(text, fontFace,
fontScale, thickness, &baseline);
    baseline += thickness;

    // Adjust the coords for left/right-justified
    // or top/bottom-justified.
    if (coord.y >= 0) {
        // Coordinates are for the top-left corner
        // of the text from the top-left of the image, so move
        // down by one row.
        coord.y += textSize.height;
    }
    else {
        // Coordinates are for the bottom-left
        // corner of the text from the bottom-left of the
        // image, so come up from the bottom.
        coord.y += img.rows - baseline + 1;
    }
    // Become right-justified if desired.
    if (coord.x < 0) {
        coord.x += img.cols - textSize.width + 1;
    }

    // Get the bounding box around the text.
    Rect boundingRect = Rect(coord.x, coord.y -
textSize.height, textSize.width, baseline +
textSize.height);

    // Draw anti-aliased text.
    putText(img, text, coord, fontFace, fontScale,
color, thickness, CV_AA);

    // Let the user know how big their text is, in
    // case they want to arrange things.
    return boundingRect;
}

// Draw a GUI button into the image, using
drawString().
// Can specify a minWidth if you want several
// buttons to all have the same width.
// Returns the bounding rect around the drawn
// button, allowing you to position buttons next to
// each other.
Rect drawButton(Mat img, string text, Point coord,
int minWidth = 0)
{
    int B = BORDER;
    Point textCoord = Point(coord.x + B, coord.y +
B);
    // Get the bounding box around the text.
    Rect rcText = drawString(img, text, textCoord,
CV_RGB(0,0,0));
    // Draw a filled rectangle around the text.
    Rect rcButton = Rect(rcText.x - B, rcText.y -
B, rcText.width + 2*B, rcText.height + 2*B);
    // Set a minimum button width.
    if (rcButton.width < minWidth)
        rcButton.width = minWidth;
    // Make a semi-transparent white rectangle.
    Mat matButton = img(rcButton);
    matButton += CV_RGB(90, 90, 90);
    // Draw a non-transparent white border.
    rectangle(img, rcButton, CV_RGB(200,200,200),
1, CV_AA);

    // Draw the actual text that will be displayed,
    // using anti-aliasing.
    drawString(img, text, textCoord,
CV_RGB(10,55,20));

    return rcButton;
}

bool isPointInRect(const Point pt, const Rect rc)
{
    if (pt.x >= rc.x && pt.x <= (rc.x + rc.width -
1))
        if (pt.y >= rc.y && pt.y <= (rc.y +
rc.height - 1))
            return true;

    return false;
}
char file[10][25][64];

void readImage(char *name){

```

```

    int i;

    for(i = 0; i < 35; i++){
        sprintf(file[0][i],"/home/ubuntu/data/%s/%s_%d.jpg",
        name, name, i);
        cout << file[0][i] << endl;
    }

}

void readALL(){

    // readImage("Joaquim",0);
    // readImage("Lucas",0);
    // readImage("Nune",1);
    // readImage("Rafael",2);
    // readImage("Robson",3);
    // readImage(Rafael,3);

}

void printfiles(){
    int i,j;
    for(i = 0; i < 4; i++){
        for(j = 0; j < 25; j++){
            cout << file[i][j] << endl;
        }
    }
}

int generateReport(DataReport report,
vector<DataStudent> student){
    ofstream myfile;
    int i;

    char *pch;
    string disciplina;
    char filename[64];

    disciplina = report.disciplina;

    date_fd = popen("date +\"%H:%M\"", "r");
fgets(current_hour, sizeof(current_hour), date_fd);

sprintf(filename,"report %s_%s.%s.csv",disciplina.c_
_str(), report.turma.c_str(),current_hour);

myfile.open (filename);
const char *bom = "\xEF\xBB\xBF";
myfile << bom;

myfile << "Relatório gerado:" << current_hour
<< endl;
myfile << "Disciplina:" << report.disciplina
<< endl;
myfile << "Turma: " << report.turma << endl;
myfile << "Professor Responsável:" <<
report.professor << endl;
myfile << "Início: " << report.startHour <<
", Fim: " << report.endHour << endl;
myfile << endl;
myfile << "Aluno, Nro. USP" << endl;

for(i = 0; i < student.size(); i ++){
    myfile << student[i].name <<
student[i].lastName << "," << student[i].numUSP <<
endl;
}
myfile << endl;
myfile << "Total:" << i << endl;

myfile.close();
return 1;
}

```

```

// Main loop that runs forever, until the user hits
Escape to quit.
void recognizeAndTrainUsingWebcam(CascadeClassifier
&faceCascade, CascadeClassifier &eyeCascade1,
CascadeClassifier &eyeCascade2)
{
    Ptr<FaceRecognizer> model;
    // vector<Mat> preprocessedFaces;
    // vector<int> faceLabels;
    Mat old_preprocessedFace;
    double old_time = 0;

    // Since we have already initialized
    // everything, lets start in Detection mode.
    m_mode = MODE_STARTUP;
    char keypress;
    Mat cameraFrame;
    Mat displayedFrame;
    string outputStr = "Unknown";
    char * fifoOutput = "/tmp/fifoOutput";
    char * fifoCollect = "/tmp/fifoCollect";
    char filename[64];
    char last_filename[64] = "";
    double first_time;
    double last_time;
    double diff_time;
    ofstream myfile;
    int identity = -1;
    bool gotFaceAndEyes = false;

    myfile.open
    ("~/home/ubuntu/victor/NewFaceRec/output.csv");
    const char *bom = "\xef\xbb\xbf";
    myfile << bom;

    int fd_output, fd_collect;
    fd_output = open(fifoOutput, O_RDWR);
    fd_collect = open(fifoCollect, O_RDWR);
    if(fd_output < 0 || fd_collect < 0){
        cout << "error to open fifo " << fifoOutput
    << endl;
    }

    // readALL();
    // printfiles();

    int localCounter = 0;
    int addCounter = 0;

    // Run forever, until the user hits Escape to
    "break" out of this loop.
    while (true) {
        first_time = (double)getTickCount();

        if(m_mode == MODE_IDLE){
            //nothing
        }else{
            cout << " Current MODE: " << m_mode
        << endl;
            if(localCounter > 3 && m_mode ==
            MODE_COLLECT_FACES ){ //discard this image
                countPickImage++;
                localCounter = 0;
                addCounter++;
            }
        }

        try{
            //to collect faces the opencv
            //will capture from camera device
            if(!imgfile){
                Mat cameraFrame;
                videoCapture >>
                cameraFrame;
                if(
                cameraFrame.empty() ) {
                    cerr << "ERROR:
                    Couldn't grab the next camera frame." << endl;
                    exit(1);
                }
                // Get a copy of the
                // camera frame that we can draw onto.
                cameraFrame.copyTo(displayedFrame);
            }
        }
    }
}

```

```

    }else{
        if(img)
            strcpy(filename,fileImage);
        else
            strcpy(filename,file[0][countPickImage]);

        cout << "file name: "
        << filename << " counter " << countPickImage <<
        endl;

        Mat image =
        imread(filename,CV_LOAD_IMAGE_COLOR);

        if (image.empty())
        {
            cout << "!!!";
            Failed imread(): image not found" << std::endl;
            // don't let the
            execution continue, else imshow() will crash.
        }

        displayedFrame.release();

        image.copyTo(displayedFrame);
        image.release();
    }
    }catch(cv::Exception &e){
        cout << e.msg << endl;
    }
}

// otherwise will get images from
// disk captured and saved by mjpg streamer (this was
// done because mjpg_Streamer is used to stream videos
// to web)
// }else{
//     char filename[64] =
    "/home/ubuntu/mjpg-streamer/output9.jpg";

    // Mat image =
    imread(filename,CV_LOAD_IMAGE_COLOR);

    // if (image.empty())
    // {
    //     cout << "!!! Failed imread():
    image not found" << std::endl;
    //     // don't let the execution
    continue, else imshow() will crash.
    // }
    // displayedFrame.release();
    // image.copyTo(displayedFrame);
    // image.release();

    // cout << " Parte 11 " << endl;
    // date_fd = popen("date
    +"%H:%M\%", "r");
    //
    fgets(current_hour,sizeof(current_hour),date_fd);
    // cout << " Parte 12 " << endl;
    // //checking if current hour =
    start/end hour of report
    // if(report.size() != 0){
    //     cout << " Parte 13 " <<
    report.size() << endl;
    //
    if(strcmp(current_hour,report[0].startHour.c_str())
    == 0 && !reporting){ //current hour = start hour of
    relatorio
    // cout << endl << "Starting
    Report " << endl ;
    // personIdentified.clear();
    // reporting = true;

    // }else
    if(strcmp(current_hour,report[0].endHour.c_str())
    == 0 && reporting){ //current hour = end hour of
    relatorio
    // reporting = false;
    // //
    vector<studentData>::iterator j;
    // cout << "Finishing Report "
    << endl;
}

```

```

        // generateReport(report[0],personIdentified);
        //           // for(j =
personIdentified.begin(); j <
personIdentified.end(); j++){
        //           // cout <<"::: " <<
(*j).name << " " << (*j).lastName << endl;
        //           // }

        //           }

        //           }

        // Run the face recognition system on
        // the camera image. It will draw some things onto the
        // given image, so make sure it is not read-only
        // memory!
        identity = -1;

        // Find a face and preprocess it to
        // have a standard size and contrast & brightness.
        Rect faceRect; // Position of detected
        face.
        Rect searchedLeftEye, searchedRightEye;
        // top-left and top-right regions of the face,
        // where eyes were searched.
        Point leftEye, rightEye; // Position
        of the detected eyes.
        Mat preprocessedFace =
        getPreprocessedFace(displayedFrame, faceWidth,
        faceCascade, eyeCascade1, eyeCascade2,
        preprocessLeftAndRightSeparately, &faceRect,
        &leftEye, &rightEye, &searchedLeftEye,
        &searchedRightEye);

        gotFaceAndEyes = false;
        if (preprocessedFace.data)
            gotFaceAndEyes = true;

        // Draw an anti-aliased rectangle
        around the detected face.
        if (faceRect.width > 0) {
            rectangle(displayedFrame, faceRect,
            CV_RGB(255, 255, 0), 2, CV_AA);

            // Draw light-blue anti-aliased
            circles for the 2 eyes.
            Scalar eyeColor =
            CV_RGB(0,255,255);
            if (leftEye.x >= 0) { // Check if
            the eye was detected
                circle(displayedFrame,
                Point(faceRect.x + leftEye.x, faceRect.y +
                leftEye.y), 6, eyeColor, 1, CV_AA);
            }
            if (rightEye.x >= 0) { // Check
            if the eye was detected
                circle(displayedFrame,
                Point(faceRect.x + rightEye.x, faceRect.y +
                rightEye.y), 6, eyeColor, 1, CV_AA);
            }
        }
        if (m_mode == MODE_STARTUP) {
            // load xml file
            loadTrainingData();
            loadClassData();

            cout << "SIZEEE " << report.size()
<< endl;

            m_mode = MODE_TRAINING;
            // m_mode = MODE_DETECTION;
        }

        else if (m_mode == MODE_DETECTION) {
            // Don't do anything special.

        }
        else if (m_mode == MODE_COLLECT_FACES)
        {
            // Check if we have detected a
            face.
            if (gotFaceAndEyes) {

```

```

                // cout << "[debug] - last
                faces " << m_latestFaces.size() << " selected
                person " << m_selectedPerson << endl;
                // Check if this face looks
                somewhat different from the previously collected
                face.
                double imageDiff =
                10000000000.0;
                if
                (old_prepprocessedFace.data) {
                    imageDiff =
                    getSimilarity(preprocessedFace,
                    old_prepprocessedFace);
                }

                // Also record when it
                happened.
                double current_time =
                (double)getTickCount();
                double timeDiff_seconds =
                (current_time - old_time)/getTickFrequency();

                // Only process the face if it
                is noticeably different from the previous frame and
                there has been noticeable time gap.
                if ((imageDiff >
                CHANGE_IN_IMAGE_FOR_COLLECTION) &&
                (timeDiff_seconds >
                CHANGE_IN_SECONDS_FOR_COLLECTION)) {
                    // Also add the mirror
                    image to the training set, so we have more training
                    data, as well as to deal with faces looking to the
                    left or right.
                    Mat mirroredFace;
                    flip(preprocessedFace,
                    mirroredFace, 1);

                    // Add the face images to
                    the list of detected faces.
                    //
                    studentlocal.Faces.push_back(preprocessedFace);
                    //
                    studentlocal.Faces.push_back(mirroredFace);
                    //
                    studentlocal.Labels.push_back(m_selectedPerson);
                    //
                    studentlocal.Labels.push_back(m_selectedPerson);

                    preprocessedFaces.push_back(preprocessedFace);
                    preprocessedFaces.push_back(mirroredFace);

                    faceLabels.push_back(m_selectedPerson);
                    faceLabels.push_back(m_selectedPerson);

                    // char aux1[64];
                    // char aux2[64];
                    // sprintf(aux2,"mkdir
                    /home/ubuntu/data/%s",studentsDB[m_selectedPerson].name.c_str());
                    // system(aux2);
                    //
                    sprintf(aux1,"/home/ubuntu/data/%s/%s_%d.jpg",stude
                    ntDB[m_selectedPerson].name.c_str(),studentsDB[m_s
                    electedPerson].name.c_str(),countPickImage);
                    //
                    imwrite(aux1,cameraFrame);
                    // Keep a reference to the
                    latest face of each person.

                    m_latestFaces[m_selectedPerson] =
                    preprocessedFaces.size() - 2; // Point to the non-
                    mirrored face.
                    // Show the number of
                    collected faces. But since we also store mirrored
                    faces, just show how many the user thinks they
                    stored.

                    countPickImage++;
                    localCounter = 0;
                    cout << "Saved face " <<
                    countPickImage - addCounter << " for person " <<
                    m_selectedPerson << endl;

```

```

        //writing message back for
web page
        char fifomessage[64];
        sprintf(fifomessage,"%d-
",countPickImage);
        //send the number of
collect to be read by php page
        pthread_mutex_lock(
&mutex_fifoOutput );
write(fd_collect,fifomessage,sizeof(char)*strlen(fi
fomessage));
        pthread_mutex_unlock(
&mutex_fifoOutput );

        // Make a white flash on
the face, so the user knows a photo has been taken.
        // Mat displayedFaceRegion
= displayedFrame(faceRect);
        // displayedFaceRegion +=
CV_RGB(90,90,90);

        // Keep a copy of the
processed face, to compare on next iteration.
        old_preprocessedFace =
preprocessedFace;
        old_time = current_time;
    }
}
else if (m_mode == MODE_TRAINING) {
    // Check if there is enough data to
train from. For Eigenfaces, we can learn just one
person if we want, but for Fisherfaces,
    // we need atleast 2 people
otherwise it will crash!
    bool haveEnoughData = true;
    if (strcmp(facerecAlgorithm,
"FaceRecognizer.Fisherfaces") == 0) {
        if ((m_numPersons < 2) ||
(m_numPersons == 2 && m_latestFaces[1] < 0) ) {
            cout << "Warning:
Fisherfaces needs atleast 2 people, otherwise there
is nothing to differentiate! Collect more data ...
<< endl;
            haveEnoughData = false;
        }
        if (m_numPersons < 1 ||
preprocessedFaces.size() <= 0 ||
preprocessedFaces.size() != faceLabels.size()) {
            cout << "Warning: Need some
training data before it can be learnt! Collect more
data ..." << endl;
            haveEnoughData = false;
        }
        if (haveEnoughData) {
            // Start training from the
collected faces using Eigenfaces or a similar
algorithm.
            model =
learnCollectedFaces(preprocessedFaces, faceLabels,
facerecAlgorithm);
            // storeTrainingData();
            // Show the internal face
recognition data, to help debugging.
            if (m_debug)
showTrainingDebugData(model, faceWidth,
faceHeight);
            //save in xml file
            // Now that training is over,
we can start recognizing!
            m_mode = MODE_RECOGNITION;
        }
    else {
        // Since there isn't enough
training data, go back to the face collection mode!
        m_mode = MODE_IDLE; //wait
command from WEB
    }
}
else if (m_mode == MODE_RECOGNITION) {

```

```

    if (gotFaceAndEyes &&
(preprocessedFaces.size() > 0) &&
(preprocessedFaces.size() == faceLabels.size())) {

        // Generate a face
approximation by back-projecting the eigenvectors &
eigenvalues.
        Mat reconstructedFace;
        reconstructedFace =
reconstructFace(model, preprocessedFace);
        if (m_debug)
            if (reconstructedFace.data)

        saveFloatMat("reconstructedFace",
&reconstructedFace);
        // Verify whether the
reconstructed face looks like the preprocessed
face, otherwise it is probably an unknown person.
        double similarity =
getSimilarity(preprocessedFace, reconstructedFace);

        imwrite("/var/www/html/reconstructed.jpg",reconstructedFace);

        imwrite("/var/www/html/preprocessedFace.jpg",reconstructedFace);

        if (similarity <
UNKNOWN_PERSON_THRESHOLD) {
            // Identify who the person
is in the preprocessed face image.
            identity = model-
>predict(preprocessedFace);
            lastPersonName = outputStr;
            // outputStr =
personNames[identity];
            // char auxout[128];
            // sprintf(auxout,"%s %s
%s",studentsDB[identity].name,studentsDB[identity].lastName,studentsDB[identity].numUSP);

            string auxout;
            auxout.append(studentsDB[identity].name);
            auxout.append(" ");
            auxout.append(studentsDB[identity].lastName);
            auxout.append(" ");
            auxout.append(studentsDB[identity].numUSP);
            outputStr = auxout;

            cout << "person Identified:
" << studentsDB[identity].name << " " <<
studentsDB[identity].lastName << endl;
            if(reporting){ //make a
report
                //check if the
identified person there is not in the string array
                DataStudent std =
DataStudent(studentsDB[identity].name,
studentsDB[identity].lastName,
studentsDB[identity].numUSP);
                int ret_ =
checkStudent(std);
                if(ret_ < 0){
                    cout << endl
<< "Adding new person for report: " << outputStr <<
endl << endl;
                    //add to the array
                }
                personIdentified.push_back(std);
            }
        }
        else {
            // Since the confidence is
low, assume it is an unknown person.
            outputStr = "Unknown";
        }
        cout << "Identity: " <<
identity << " Name: " << outputStr.c_str() << " .
Similarity: " << similarity << endl;
    }
}

```

```

        if(img &&
(strcmp(filename,last_filename) != 0))
            myfile << "file," <<
filename << ", identity," << outputStr << ",
similarity," << similarity << ",";
            char fifomessage[64];
            sprintf(fifomessage,"%s-
",outputStr.c_str());
            //send last person recognized
            to fifo output to be read by php page
            pthread_mutex_lock(
&mutex_fifoOutput );
}

while(write(fd_output,fifomessage,sizeof(char)*strlen(fifomessage)) < 0);
            pthread_mutex_unlock(
&mutex_fifoOutput );
            // Show the confidence rating
            for the recognition in the mid-top of the display.
            int cx = (displayedFrame.cols -
faceWidth) / 2;
            Point ptBottomRight = Point(cx
- 5, BORDER + faceHeight);
            Point ptTopLeft = Point(cx -
15, BORDER);
            // Draw a gray line showing the
            threshold for an "unknown" person.
            Point ptThreshold =
Point(ptTopLeft.x, ptBottomRight.y - (1.0 -
UNKNOWN_PERSON_THRESHOLD) * faceHeight);
            rectangle(displayedFrame,
ptThreshold, Point(ptBottomRight.x, ptThreshold.y),
CV_RGB(200,200,200), 1, CV_AA);
            // Crop the confidence rating
            between 0.0 to 1.0, to show in the bar.
            double confidenceRatio = 1.0 -
min(max(similarity, 0.0), 1.0);
            Point ptConfidence =
Point(ptTopLeft.x, ptBottomRight.y -
confidenceRatio * faceHeight);
            // Show the light-blue
            confidence bar.
            rectangle(displayedFrame,
ptConfidence, ptBottomRight, CV_RGB(0,255,255),
CV_FILLED, CV_AA);
            // Show the gray border of the
            bar.
            rectangle(displayedFrame,
ptTopLeft, ptBottomRight, CV_RGB(200,200,200), 1,
CV_AA);

        }
        // else if (m_mode == MODE_DELETE_ALL)
{
        //     // Restart everything!
        //     m_selectedPerson = -1;
        //     m_numPersons = 0;
        //     m_latestFaces.clear();
        //     preprocessedFaces.clear();
        //     faceLabels.clear();
        //     old_preprocessedFace = Mat();

        //     // Restart in Detection mode.
        //     m_mode = MODE_DETECTION;
        // }
        else if(m_mode == MODE_EXIT){
            cout << "Exiting mode" << endl;
            myfile.close();
            exit(1);

        }else{
            cerr << "ERROR: Invalid run mode "
<< m_mode << endl;
            exit(1);
        }

        // Show the current preprocessed face
        // in the top-center of the display.
        int cx = (displayedFrame.cols -
faceWidth) / 2;
        if (preprocessedFace.data) {
            // Get a BGR version of the face,
            since the output is BGR color.
            Mat srcBGR =
Mat(preprocessedFace.size(), CV_8UC3);

```

```

        cvtColor(preprocessedFace, srcBGR,
CV_GRAY2BGR);
        // Get the destination ROI (and
        // make sure it is within the image!).
        //min(m_gui faces top + i *
faceHeight, displayedFrame.rows - faceHeight);
        Rect dstRC = Rect(cx, BORDER,
faceWidth, faceHeight);
        Mat dstROI = displayedFrame(dstRC);
        // Copy the pixels from src to dst.
        srcBGR.copyTo(dstROI);
    }

imwrite("/var/www/html/detected.jpg",displayedFrame
);

            // IMPORTANT: Wait for atleast 20
            milliseconds, so that the image can be displayed on
            the screen!
            // Also checks if a key was pressed in
            the GUI window. Note that it should be a "char" to
            support Linux.
            if(countPickImage == (NUMBER_COLLECT +
addCounter)){
                printf("25 images were collected,
start trainning \n");
                countPickImage = 0; //default value
                addCounter = 0;
                //save faces for each student
                // studentlocal.Faces =
preprocessedFaces;
                // studentlocal.Labels =
faceLabels;

                //
                studentsDB.push_back(studentlocal);

                // vector<Mat>::iterator it;
                // for(it =
studentlocal.Faces.begin(); it !=
studentlocal.Faces.end(); it++){
                    //
                    preprocessedFaces.push_back(*it);
                    //
                }

                // vector<int>::iterator it_label;
                // for(it_label =
studentlocal.Labels.begin(); it_label !=
studentlocal.Labels.end(); it_label++){
                    //
                    faceLabels.push_back(*it_label);
                    //
                }

                //store data in xml file
                storeTrainingData();
                m_mode = MODE_TRAINING;
                // m_mode = MODE_DETECTION;
            }
            if(imgfile)
                localCounter++;

        }

        last_time = (double)getTickCount();
        diff_time = (last_time -
first_time)/getTickFrequency();

        cout << "Time Elapsed: " << diff_time <<
endl;

        if(img && (strcmp(filename,last_filename)
!= 0) && gotFaceAndEyes)
            myfile << "Time Elapsed," << diff_time
<< endl;

        strcpy(last_filename,filename);

    } //end while
}

```

```

#define MAX_BUF 1024

void *get_message_web(void *ptr){
    int fd1,fd2;
    char * fifoRead = "/tmp/fifoRead";
    char * fifoWrite = "/tmp/fifoWrite";

    char message[256];
    int i,ret;
    char *pch;

    fd1 = open(fifoWrite,O_RDWR);
    fd2 = open(fifoRead, O_RDWR);

    if(fd1 < 0 || fd2 < 0){
        cout << "error to open fifo" << endl;
        exit(0);
    }
    while(1){

        while(read(fd2, message, MAX_BUF) < 0);
        // while(write(fd2,"Message received ",sizeof("Message received "))<0); //send ack
        cout << "Message received " << message << endl;

        pch = strtok (message,"-");
        cout << "PCH " << pch << endl;
        i = 0;
        if(strcmp(pch,"collect") == 0){
            while (pch != NULL && i < 4){
                pch = strtok (NULL, "-");
                strcpy(collect[i++],pch);
            }

            char name[64];
            char id[2];
            sprintf(id,"%d",m_numPersons);
            /
strcpy(studentlocal.name,collect[0]);
            /
strcpy(studentlocal.lastName,collect[1]);
            /
strcpy(studentlocal.numUSP,collect[2]);
            /
sprintf(studentlocal.id,"%d",m_numPersons);
            cout << "SIZEEE " << report.size() << endl;

            DataStudent studentlocal =
DataStudent(collect[0],collect[1],collect[2]);

            cout << studentlocal.name <<
studentlocal.lastName << studentlocal.numUSP <<
endl;
            sprintf (name,"%s %s",collect[0],collect[1],collect[2]);
            printf("Data Received: %s\n",name);

            cout << "vector size " <<
studentsDB.size() << " Empty " <<
studentsDB.empty() << endl;
            // studentsDB.resize(m_numPersons+1);
            ret = checkStudent(studentlocal);
            if(ret < 0){
                studentsDB.push_back(studentlocal);
                countPickImage = 0; //reset number
of images captured
                readImage(collect[0]);

                // idl++;
                cout << "Incrementing number of
persons, number persons was = " << m_numPersons <<
endl;
                cout << "[debug] - last faces " <<
m_latestFaces.size() << " selected person " <<
m_selectedPerson << endl;
                // Check if there is already a
person without any collected faces, then use that
person instead.
                // This can be checked by seeing if
an image exists in their "latest collected face".
                if ((m_numPersons == 0) ||
(m_latestFaces[m_numPersons-1] >= 0)) {
                    // Add a new person.

```

```

                    m_numPersons++;
                    personNames.push_back(name);
                    m_latestFaces.push_back(-1); // Allocate space for an extra person.
                    cout << "Current Num Persons: " << m_numPersons << endl;
                }
                // Use the newly added person. Also
use the newest person even if that person was
empty.
                m_selectedPerson = m_numPersons -
1;
                // system("service streamcam
stop");

                // // Get access to the webcam.
                // initWebcam();
                imgfile = true;
                img = false;
                pthread_mutex_lock( &mutex_web );
                m_mode = MODE_COLLECT_FACES;
                pthread_mutex_unlock( &mutex_web );
                printf("Starting collect faces\n");
                cout << "SIZEEE " << report.size() << endl;
            } else{
                cout << "Failed to add person " <<
endl;
                pthread_mutex_lock( &mutex_web );
                m_mode = MODE_RECOGNITION;
                pthread_mutex_unlock( &mutex_web );
            }
        } //received a message to collect faces;
        else if(strcmp(pch,"recognition") == 0){
            pthread_mutex_lock( &mutex_web );
            m_mode = MODE_RECOGNITION;
            imgfile = false;
            pthread_mutex_unlock( &mutex_web );
        } //recognition mode
        else if(strcmp(pch,"training") == 0){
            pthread_mutex_lock( &mutex_web );
            m_mode = MODE_TRAINING;
            pthread_mutex_unlock( &mutex_web );
        } else if(strcmp(pch,"detection") == 0){
            pthread_mutex_lock( &mutex_web );
            m_mode = MODE_DETECTION;
            pthread_mutex_unlock( &mutex_web );
        } else if(strcmp(pch,"relatorio") == 0){
            pthread_mutex_lock( &mutex_web );
            m_mode = MODE_IDLE;
            pthread_mutex_unlock( &mutex_web );
            while (pch != NULL && i < 5){

                pch = strtok (NULL, "-");
                strcpy(relatorio[i++],pch);
            }

            DataReport reportLocal =
DataReport(relatorio[0],relatorio[1],relatorio[2],r
elatorio[3],relatorio[4]);

            printData();
            if(checkReport(reportLocal) < 0){
//there is no report with the same class and number
                report.push_back(reportLocal);
                numRelatorios++;
                storeClassData();
            } else{
                cout << "report is already present
in DB" << endl;
            }
        }

        pthread_mutex_lock( &mutex_web );
        m_mode = MODE_RECOGNITION;
        pthread_mutex_unlock( &mutex_web );
    } else if(strcmp(pch,"iniciar") == 0){
        pthread_mutex_lock( &mutex_web );
        m_mode = MODE_DETECTION;
        pthread_mutex_unlock( &mutex_web );
        system("service streamcam start");

    } else if(strcmp(pch,"fechar") == 0){
        pthread_mutex_lock( &mutex_web );
        m_mode = MODE_EXIT;
        pthread_mutex_unlock( &mutex_web );
        system("service streamcam stop");
    }
}

```

```

}else if(strcmp(pch,"removeStudent") == 0){
    pthread_mutex_lock( &mutex_web );
    m_mode = MODE_IDLE;
    pthread_mutex_unlock( &mutex_web );
    pch = strtok (NULL, "-"); //pch gets
numUSP

    ret = removeStudent(pch);
    if(ret){
        cout << "Student removed " << endl;
    }else{
        cout << "Student is not present" <<
endl;
    }

    if(m_numPersons == 0){
        pthread_mutex_lock( &mutex_web );
        m_mode = MODE_IDLE;
        pthread_mutex_unlock( &mutex_web );
        cout << "Student DB is empty,
stopping recognition" << endl;
    }
    else{
        pthread_mutex_lock( &mutex_web );
        cout << "retraining" << endl;
        m_mode = MODE_TRAINING;
        pthread_mutex_unlock( &mutex_web );
    }

}else if(strcmp(pch,"removeReport") == 0){
    pthread_mutex_lock( &mutex_web );
    m_mode = MODE_IDLE;
    pthread_mutex_unlock( &mutex_web );
    pch = strtok (NULL, "-"); //pch gets
disciplina

    string disciplina = pch;
    cout << "pch " << pch << " ds " <<
disciplina << endl;
    pch = strtok (NULL, "-"); //pch gets
turma

    cout << "pch " << pch << endl;

    ret = removeReport(disciplina,pch);
    if(ret){
        cout << "report removed " << endl;
    }else{
        cout << "report is not present" <<
endl;
    }

    if(numRelatorios == 0)
        cout << "report is empty" << endl;
    pthread_mutex_lock( &mutex_web );
    if(m_numPersons > 0)
        m_mode = MODE_RECOGNITION;
    else{
        cout << "Waiting new data to start
training" << endl;
        m_mode = MODE_TRAINING;
    }
    pthread_mutex_unlock( &mutex_web );
}else if(strcmp(pch,"image") == 0){

    pthread_mutex_lock( &mutex_web );
    pch = strtok (NULL, "-"); //pch gets
disciplina

    strcpy(fileImage,pch);
    cout << "File name " << fileImage <<
endl;
    imgfile = true;
    img = true;
    pthread_mutex_unlock( &mutex_web );
}

```

```

}
sleep(1);

}
close(fd1);
close(fd2);
}

int main(int argc, char *argv[])
{
    CascadeClassifier faceCascade;
    CascadeClassifier eyeCascadel;
    CascadeClassifier eyeCascade2;
    pthread_t thread1, thread_get_message_web;
    int iret1, iret2;
    const char *message1 = "Thread 1";
    cout << "WebcamFaceRec, by Shervin Emami
(www.shervinemani.info), June 2012." << endl;
    cout << "Enhanced for beaglebone Black by
Victor Nascimento in June 2015" << endl;
    cout << "Realtime face detection + face
recognition from a webcam using LBP and Eigenfaces
or Fisherfaces." << endl;
    cout << "Compiled with OpenCV version " <<
CV_VERSION << endl << endl;
    cout << "\nMenu options: \n\t 0 - Collect Faces
\n\t 1 - Recognition \n\t 2 - Training \n\t 3 -
Detection (Default) \n\t 4 - Load xml database \n\t
5 - Store xml database \n\t 6 - load image file \n"
<< endl;

// date_fd = popen("date +\"%H:%M\"","r");
// fgets(current_hour,sizeof(current_hour),date_fd);

printf("Program started at:
%s\n",current_hour);
report.clear();
personIdentified.clear();
preprocessedFaces.clear();
faceLabels.clear();
initWebcam();

// iret1 = pthread_create( &thread1, NULL,
changeMode, (void*) message1);
iret2 =
pthread_create(&thread_get_message_web,NULL,get_mes
sage_web,NULL);
if(iret2)
{
    fprintf(stderr,"Error - pthread_create()
return code: %d\n",iret2);
    exit(EXIT_FAILURE);
}

// Load the face and 1 or 2 eye detection XML
classifiers.
initDetectors(faceCascade, eyeCascadel,
eyeCascade2);

cout << endl;

// Run Face Recognition interactively from the
webcam. This function runs until the user quits.
recognizeAndTrainUsingWebcam(faceCascade,
eyeCascadel, eyeCascade2);

// pclose(date_fd);
return 0;
}

```

APÊNDICE C - Tabela dados do processamento da *BeagleBone Black*

Tempo(H:M:S)	%CPU	%Memória	%Frequência	Frequência(MHz)
03:28:35	99.00%	44.69%	100.00%	1000
03:28:37	99.00%	44.69%	100.00%	1000
03:28:39	99.00%	44.69%	100.00%	1000
03:28:41	99.00%	44.69%	100.00%	1000
03:28:43	99.00%	44.69%	100.00%	1000
03:28:45	99.00%	44.69%	100.00%	1000
03:28:47	98.99%	44.69%	100.00%	1000
03:28:49	99.00%	44.69%	100.00%	1000
03:28:51	99.00%	44.69%	100.00%	1000
03:28:53	99.00%	44.69%	100.00%	1000
03:40:34	99.65%	50.53%	100.00%	1000
03:40:36	84.00%	50.56%	90.96%	909.55
03:40:38	97.50%	50.56%	100.00%	1000
03:40:40	95.98%	50.56%	100.00%	1000
03:40:42	68.21%	50.58%	81.20%	812
03:40:44	87.00%	50.58%	89.20%	892.04
03:40:46	98.00%	50.58%	100.00%	1000
03:40:48	96.97%	50.58%	100.00%	1000
03:40:50	97.51%	50.58%	100.00%	1000
03:40:52	97.50%	50.61%	100.00%	1000
03:40:54	96.50%	50.61%	100.00%	1000
03:40:56	82.32%	50.61%	94.85%	948.51
03:40:58	85.93%	50.61%	89.95%	899.5
03:41:00	97.49%	50.61%	100.00%	1000
03:41:02	97.51%	50.61%	100.00%	1000
03:41:04	97.99%	50.61%	100.00%	1000
03:41:06	97.01%	50.63%	100.00%	1000
03:41:08	98.49%	50.63%	100.00%	1000
03:41:10	72.96%	50.63%	86.16%	861.62
03:41:12	96.50%	50.63%	100.00%	1000
03:41:14	96.00%	50.63%	100.00%	1000

03:41:16	97.01%	50.63%	100.00%	1000
03:41:18	97.99%	50.63%	100.00%	1000
03:41:20	97.49%	50.63%	100.00%	1000
03:41:22	97.00%	50.63%	100.00%	1000
03:41:24	96.52%	50.63%	100.00%	1000
03:41:26	97.49%	50.65%	100.00%	1000
03:41:28	97.51%	50.65%	100.00%	1000
03:41:30	96.98%	50.65%	100.00%	1000
03:41:32	75.76%	50.65%	92.48%	924.75
03:41:34	97.50%	50.65%	98.49%	984.92
03:41:36	76.02%	50.65%	83.23%	832.34
03:41:38	97.50%	50.65%	100.00%	1000
03:41:40	83.33%	50.65%	90.45%	904.48
03:41:42	93.97%	50.65%	91.40%	914
03:41:44	97.00%	50.68%	100.00%	1000
03:41:46	95.50%	50.68%	100.00%	1000
03:41:48	97.49%	50.68%	100.00%	1000
03:41:50	96.02%	50.68%	100.00%	1000
03:41:52	97.99%	50.68%	100.00%	1000
03:41:54	90.95%	50.68%	99.90%	998.99
03:41:56	72.08%	50.68%	83.28%	832.84
03:41:58	97.50%	50.68%	98.79%	987.94
03:42:00	97.00%	50.70%	100.00%	1000
03:42:02	93.50%	50.78%	100.00%	1000
03:42:04	97.50%	50.78%	100.00%	1000
03:42:06	96.98%	50.78%	100.00%	1000
03:42:08	96.52%	50.78%	100.00%	1000
03:42:10	96.98%	50.78%	100.00%	1000
03:42:12	76.65%	50.78%	79.00%	790
03:42:14	97.50%	50.78%	100.00%	1000
Média	93.94%	49.67%	97.53%	975.30
Desvio Padrão	7.76%	2.21%	5.32%	53.19

APÊNDICE D - Tabelas de testes do algoritmo de detecção com Haar Cascade

Tabela 1: Threshold = 0.5

Arquivo de Entrada	Nome Identificado	Similaridade	Tempo(s)	Resultado
Robson_1.jpg	Robson Silva 1111114	0.349555	1.54155	Sucesso
Robson_2.jpg	Robson Silva 1111114	0.440305	1.5488	Sucesso
Robson_4.jpg	Robson Silva 1111114	0.370024	1.54993	Sucesso
Robson_5.jpg	Robson Silva 1111114	0.415198	1.63363	Sucesso
Robson_7.jpg	Robson Silva 1111114	0.377051	1.60524	Sucesso
Robson_8.jpg	Robson Silva 1111114	0.426066	1.58762	Sucesso
Robson_10.jpg	Unknown	0.521566	1.25649	Falhou
Robson_11.jpg	Robson Silva 1111114	0.351324	1.56552	Sucesso
Robson_13.jpg	Robson Silva 1111114	0.360554	1.55882	Sucesso
Robson_14.jpg	Robson Silva 1111114	0.342467	1.57138	Sucesso
Robson_16.jpg	Robson Silva 1111114	0.401532	1.69852	Sucesso
Robson_17.jpg	Robson Silva 1111114	0.365334	1.62381	Sucesso
Robson_19.jpg	Unknown	0.624439	1.23784	Falhou
Robson_20.jpg	Robson Silva 1111114	0.340411	1.60443	Sucesso
Robson_22.jpg	Unknown	0.5059791	1.23044	Falhou
Robson_23.jpg	Robson Silva 1111114	0.490274	1.57487	Sucesso
Robson_25.jpg	Robson Silva 1111114	0.444605	1.59316	Sucesso
Robson_26.jpg	Robson Silva 1111114	0.306372	1.5781	Sucesso
Robson_28.jpg	Robson Silva 1111114	0.414766	1.6003	Sucesso
Robson_29.jpg	Robson Silva 1111114	0.474255	1.55279	Sucesso
Robson_31.jpg	Robson Silva 1111114	0.346875	1.51642	Sucesso
Robson_32.jpg	Robson Silva 1111114	0.451032	1.58542	Sucesso
Robson_34.jpg	Unknown	0.5059791	1.23635	Falhou
Nune_6.jpg	Unknown	0.711003	1.82266	Falhou
Nune_8.jpg	Unknown	0.674126	1.51071	Falhou
Nune_16.jpg	Unknown	0.529269	1.22299	Falhou
Nune_19.jpg	Unknown	0.793143	1.28813	Falhou
Nune_22.jpg	Unknown	0.52475	1.31289	Falhou
Nune_25.jpg	Unknown	0.760392	1.25861	Falhou
Nune_26.jpg	Unknown	0.763784	1.24053	Falhou
Nune_27.jpg	Unknown	0.704464	1.23723	Falhou
Nune_33.jpg	Unknown	0.704727	1.26339	Falhou
Nune_34.jpg	Unknown	0.793143	1.76407	Falhou
Lucas_1.jpg	Unknown	0.57731	2.64732	Falhou
Lucas_3.jpg	Unknown	0.584095	2.09906	Falhou
Lucas_5.jpg	Unknown	0.584798	1.34113	Falhou
Lucas_6.jpg	Unknown	0.630378	1.32325	Falhou
Lucas_8.jpg	Unknown	0.670188	1.27738	Falhou
Lucas_9.jpg	Unknown	0.633048	1.32835	Falhou
Lucas_10.jpg	Unknown	0.61336	1.30471	Falhou
Lucas_11.jpg	Unknown	0.6323015	1.28922	Falhou
Lucas_13.jpg	Lucas Telles 1111111	0.492203	1.60213	Sucesso
Lucas_14.jpg	Unknown	0.588417	1.27553	Falhou
Lucas_16.jpg	Unknown	0.554883	1.31197	Falhou
Lucas_17.jpg	Lucas Telles 1111111	0.393599	1.608	Sucesso
Lucas_18.jpg	Unknown	0.625307	1.29784	Falhou
Lucas_20.jpg	Lucas Telles 1111111	0.435443	1.58815	Sucesso
Lucas_21.jpg	Lucas Telles 1111111	0.491181	1.6338	Sucesso
Lucas_23.jpg	Lucas Telles 1111111	0.438079	1.62433	Sucesso
Lucas_24.jpg	Lucas Telles 1111111	0.433368	1.60092	Sucesso
Lucas_26.jpg	Lucas Telles 1111111	0.408063	1.65198	Sucesso
Lucas_28.jpg	Lucas Telles 1111111	0.436069	1.57492	Sucesso
Lucas_29.jpg	Unknown	0.656955	2.06845	Falhou
Lucas_31.jpg	Lucas Telles 1111111	0.433651	3.20427	Sucesso
Lucas_34.jpg	Lucas Telles 1111111	0.432173	1.62696	Sucesso
Rafael_1.jpg	Unknown	0.561113	1.31367	Falhou
Rafael_2.jpg	Unknown	0.536446	1.28824	Falhou
Rafael_3.jpg	Unknown	0.637384	1.24349	Falhou
Rafael_4.jpg	Unknown	0.504188	1.28693	Falhou
Rafael_6.jpg	Unknown	0.556387	1.26282	Falhou
Rafael_7.jpg	Unknown	0.599975	1.25517	Falhou
Rafael_8.jpg	Unknown	0.551111	1.23908	Falhou
Rafael_9.jpg	Unknown	0.647049	1.29649	Falhou
Rafael_11.jpg	Unknown	0.555905	1.28494	Falhou
Rafael_12.jpg	Unknown	0.568399	1.24051	Falhou
Rafael_13.jpg	Unknown	0.577498	1.25137	Falhou
Rafael_14.jpg	Unknown	0.526031	1.43951	Falhou
Rafael_16.jpg	Unknown	0.546688	1.22654	Falhou
Rafael_17.jpg	Unknown	0.537088	1.26537	Falhou
Rafael_18.jpg	Unknown	0.548976	1.24259	Falhou
Rafael_19.jpg	Unknown	0.634625	1.44776	Falhou
Rafael_21.jpg	Unknown	0.594597	1.25319	Falhou
Rafael_22.jpg	Unknown	0.60166	1.22039	Falhou
Rafael_23.jpg	Unknown	0.598499	1.27415	Falhou
Rafael_24.jpg	Unknown	0.569201	1.28006	Falhou
Rafael_26.jpg	Unknown	0.660519	1.25969	Falhou
Rafael_27.jpg	Unknown	0.584062	1.27722	Falhou
Rafael_28.jpg	Unknown	0.660336	1.29189	Falhou
Rafael_29.jpg	Unknown	0.658116	1.29466	Falhou
Rafael_31.jpg	Unknown	0.658907	1.29886	Falhou
Rafael_32.jpg	Unknown	0.64709	1.26148	Falhou
Rafael_33.jpg	Unknown	0.599975	1.23064	Falhou
Rafael_34.jpg	Unknown	0.637384	1.29263	Falhou
Japa_1.jpg	Japa Kim 1622211	0.399707	1.65767	Sucesso
Japa_2.jpg	Japa Kim 1622211	0.423085	1.61752	Sucesso
Japa_4.jpg	Japa Kim 1622211	0.39339	1.70538	Sucesso
Japa_5.jpg	Japa Kim 1622211	0.438068	1.72809	Sucesso
Japa_7.jpg	Japa Kim 1622211	0.368878	1.63031	Sucesso
Japa_9.jpg	Japa Kim 1622211	0.38349	1.66099	Sucesso
Japa_10.jpg	Japa Kim 1622211	0.385931	1.7137	Sucesso
Japa_14.jpg	Japa Kim 1622211	0.430921	1.66937	Sucesso
Japa_15.jpg	Japa Kim 1622211	0.445857	1.70212	Sucesso
Japa_17.jpg	Japa Kim 1622211	0.483096	1.64842	Sucesso

Japa_18.jpg	Japa Kim 1622211	0.406627	1.67599	Sucesso
Japa_20.jpg	Japa Kim 1622211	0.475414	1.6691	Sucesso
Japa_22.jpg	Japa Kim 1622211	0.430822	1.70344	Sucesso
Japa_23.jpg	Unknown	0.503496	1.37818	Falhou
Japa_25.jpg	Japa Kim 1622211	0.413249	1.67776	Sucesso
Japa_26.jpg	Japa Kim 1622211	0.367067	1.68588	Sucesso
Japa_28.jpg	Japa Kim 1622211	0.423679	1.62804	Sucesso
Japa_30.jpg	Japa Kim 1622211	0.32825	1.68501	Sucesso
Japa_31.jpg	Japa Kim 1622211	0.376852	1.71893	Sucesso
JapaMan_0.jpg	JapaMan KimK 1234567	0.499529	1.6796	Sucesso
JapaMan_5.jpg	Unknown	0.568504	1.31727	Falhou
JapaMan_9.jpg	JapaMan KimK 1234567	0.472461	1.74153	Sucesso
JapaMan_11.jpg	JapaMan KimK 1234567	0.475235	1.69117	Sucesso
JapaMan_12.jpg	JapaMan KimK 1234567	0.450699	1.69634	Sucesso
JapaMan_14.jpg	Unknown	0.505556	1.31153	Falhou
JapaMan_15.jpg	Unknown	0.627536	1.28596	Falhou
JapaMan_18.jpg	JapaMan KimK 1234567	0.49724	1.70576	Sucesso
JapaMan_20.jpg	JapaMan KimK 1234567	0.445735	1.71951	Sucesso
JapaMan_21.jpg	JapaMan KimK 1234567	0.422488	1.69632	Sucesso
JapaMan_23.jpg	Unknown	0.571361	1.31957	Falhou
JapaMan_24.jpg	Unknown	0.584103	1.30559	Falhou
JapaMan_26.jpg	Unknown	0.655137	1.27257	Falhou
JapaMan_30.jpg	JapaMan KimK 1234567	0.460389	1.7037	Sucesso
JapaMan_32.jpg	JapaMan KimK 1234567	0.366202	1.67249	Sucesso
JapaChild_0.jpg	Unknown	0.606864	1.29195	Falhou
JapaChild_1.jpg	Unknown	0.533522	1.31223	Falhou
JapaChild_3.jpg	Unknown	0.606864	1.28684	Falhou
JapaChild_4.jpg	JapaChild Assaki 14132266	0.370904	1.64218	Sucesso
JapaChild_6.jpg	Unknown	0.606864	1.30819	Falhou
JapaChild_7.jpg	Unknown	0.612269	1.33976	Falhou
JapaChild_8.jpg	Unknown	0.643818	1.34523	Falhou
JapaChild_9.jpg	Unknown	0.578729	1.34355	Falhou
JapaChild_11.jpg	Unknown	0.536591	1.30722	Falhou
JapaChild_15.jpg	Unknown	0.63906	1.32619	Falhou
JapaChild_16.jpg	Unknown	0.573721	1.32395	Falhou
JapaChild_24.jpg	Unknown	0.606864	1.29084	Falhou
JapaChild_25.jpg	Unknown	0.533522	1.31442	Falhou
JapaChild_26.jpg	Unknown	0.509596	1.3266	Falhou
JapaChild_28.jpg	Unknown	0.612269	1.34089	Falhou
JapaChild_29.jpg	Unknown	0.595973	1.27889	Falhou
JapaChild_31.jpg	Unknown	0.569156	1.33429	Falhou
JapaChild_32.jpg	Unknown	0.62749	1.32243	Falhou
JapaChild_34.jpg	Unknown	0.606864	1.31296	Falhou
JapaNew_0.jpg	Unknown	0.677201	1.32514	Falhou
JapaNew_1.jpg	Unknown	0.605386	1.34186	Falhou
JapaNew_3.jpg	Unknown	0.637109	1.40861	Falhou
JapaNew_4.jpg	Unknown	0.637109	1.38734	Falhou
JapaNew_5.jpg	Unknown	0.522852	1.30995	Falhou
JapaNew_7.jpg	Unknown	0.575926	1.32556	Falhou
JapaNew_8.jpg	Unknown	0.575926	1.37348	Falhou
JapaNew_9.jpg	Unknown	0.576967	1.34131	Falhou
JapaNew_10.jpg	Unknown	0.624824	1.33501	Falhou
JapaNew_12.jpg	Unknown	0.631206	1.32303	Falhou
JapaNew_13.jpg	Unknown	0.647633	1.33039	Falhou
JapaNew_15.jpg	Unknown	0.540276	1.34342	Falhou
JapaNew_17.jpg	Unknown	0.587175	1.34434	Falhou
JapaNew_18.jpg	Unknown	0.587175	1.34595	Falhou
JapaNew_19.jpg	Unknown	0.577369	1.32855	Falhou
JapaNew_21.jpg	Unknown	0.633068	1.32443	Falhou
JapaNew_22.jpg	Unknown	0.632874	1.30925	Falhou
JapaNew_23.jpg	Unknown	0.640888	1.30133	Falhou
JapaNew_24.jpg	Unknown	0.625342	1.31512	Falhou
JapaNew_26.jpg	Unknown	0.591778	1.32809	Falhou
JapaNew_27.jpg	Unknown	0.644837	1.34464	Falhou
JapaNew_28.jpg	Unknown	0.641514	1.31331	Falhou
JapaNew_30.jpg	Unknown	0.523534	1.26571	Falhou
JapaNew_31.jpg	Unknown	0.616136	1.37477	Falhou
JapaNew_32.jpg	Unknown	0.650356	1.29985	Falhou
JapaNew_33.jpg	Unknown	0.605386	1.75588	Falhou
JapaFat_0.jpg	Unknown	0.600604	2.2792	Sucesso
JapaFat_2.jpg	Unknown	0.600604	3.28977	Sucesso
JapaFat_6.jpg	Unknown	0.617925	1.37668	Sucesso
JapaFat_7.jpg	Unknown	0.617925	1.35724	Sucesso
JapaFat_8.jpg	Unknown	0.617925	1.3621	Sucesso
JapaFat_10.jpg	Unknown	0.594339	1.31812	Sucesso
JapaFat_11.jpg	Unknown	0.594339	1.35353	Sucesso
JapaFat_12.jpg	Unknown	0.554506	1.38651	Sucesso
JapaFat_14.jpg	Unknown	0.504349	1.34458	Sucesso
JapaFat_15.jpg	Unknown	0.504349	1.31493	Sucesso
JapaFat_16.jpg	Unknown	0.504349	1.31248	Sucesso
JapaFat_18.jpg	Unknown	0.592161	1.34664	Sucesso
JapaFat_19.jpg	Unknown	0.594319	1.34568	Sucesso
JapaFat_20.jpg	Unknown	0.594319	1.36973	Sucesso
JapaFat_22.jpg	Unknown	0.51651	1.37658	Sucesso
JapaFat_23.jpg	Unknown	0.569124	1.32887	Sucesso
JapaFat_24.jpg	Unknown	0.523746	1.34122	Sucesso
JapaFat_26.jpg	JapaNew Assaki 1223226	0.461099	1.71911	Falhou
JapaFat_27.jpg	JapaNew Assaki 1223226	0.461099	1.60683	Falhou
JapaFat_29.jpg	Unknown	0.608981	1.33908	Sucesso
JapaFat_30.jpg	Unknown	0.608981	1.36363	Sucesso
JapaFat_31.jpg	Unknown	0.533811	1.39242	Sucesso
JapaFat_33.jpg	Unknown	0.601084	1.35216	Sucesso
JapaFat_34.jpg	Unknown	0.601084	1.36485	Sucesso
JapaUS_0.jpg	Unknown	0.720267	1.31653	Falhou

JapaUS_2.jpg	Unknown	0.696113	1.25684	Falhou
JapaUS_3.jpg	Unknown	0.696113	1.25743	Falhou
JapaUS_4.jpg	Unknown	0.71646	1.30213	Falhou
JapaUS_5.jpg	Unknown	0.71646	1.35064	Falhou
JapaUS_7.jpg	Unknown	0.62590	1.31178	Falhou
JapaUS_8.jpg	Unknown	0.631739	1.29152	Falhou
JapaUS_9.jpg	Unknown	0.581703	1.32346	Falhou
JapaUS_11.jpg	Unknown	0.511793	1.31156	Falhou
JapaUS_12.jpg	Unknown	0.68988	1.27752	Falhou
JapaUS_13.jpg	Unknown	0.706908	1.28637	Falhou
JapaUS_14.jpg	Unknown	0.714062	1.2956	Falhou
JapaUS_16.jpg	Unknown	0.725941	1.34023	Falhou
JapaUS_17.jpg	Unknown	0.712146	1.31337	Falhou
JapaUS_18.jpg	Unknown	0.686525	1.34248	Falhou
JapaUS_20.jpg	Unknown	0.695756	1.31528	Falhou
JapaUS_21.jpg	Unknown	0.702007	1.27787	Falhou
JapaUS_22.jpg	Unknown	0.686911	1.2973	Falhou
JapaUS_23.jpg	Unknown	0.651061	1.29777	Falhou
JapaUS_25.jpg	Unknown	0.644131	1.34218	Falhou
JapaUS_26.jpg	Unknown	0.687146	1.31285	Falhou
JapaUS_27.jpg	Unknown	0.7305039	1.31261	Falhou
JapaUS_28.jpg	Unknown	0.702007	1.29713	Falhou
JapaUS_30.jpg	Unknown	0.725941	1.31011	Falhou
JapaUS_31.jpg	Unknown	0.717032	1.27922	Falhou
JapaUS_32.jpg	Unknown	0.656526	1.32203	Falhou
JapaUS_34.jpg	Unknown	0.686911	1.31184	Falhou
JapaCN_0.jpg	JapaCN Nawan 7963226	0.455271	1.64417	Sucesso
JapaCN_1.jpg	JapaCN Nawan 7963226	0.497959	1.6547	Sucesso
JapaCN_3.jpg	Unknown	0.573734	1.29274	Falhou
JapaCN_4.jpg	JapaCN Nawan 7963226	0.463517	1.62274	Sucesso
JapaCN_6.jpg	Unknown	0.591603	1.32287	Falhou
JapaCN_7.jpg	Unknown	0.606269	1.31488	Falhou
JapaCN_8.jpg	JapaCN Nawan 7963226	0.48679	1.65063	Sucesso
JapaCN_10.jpg	Unknown	0.649111	1.33412	Falhou
JapaCN_11.jpg	Unknown	0.684515	1.37316	Falhou
JapaCN_13.jpg	Unknown	0.559955	1.33316	Falhou
JapaCN_14.jpg	JapaCN Nawan 7963226	0.49244	1.64751	Sucesso
JapaCN_16.jpg	JapaCN Nawan 7963226	0.495274	1.65331	Sucesso
JapaCN_17.jpg	JapaCN Nawan 7963226	0.46611	1.68316	Sucesso
JapaCN_19.jpg	Unknown	0.547638	1.33916	Falhou
JapaCN_20.jpg	Unknown	0.686483	1.30272	Falhou
JapaCN_21.jpg	Unknown	0.702884	1.31549	Falhou
JapaCN_23.jpg	JapaCN Nawan 7963226	0.428524	1.6325	Sucesso
JapaCN_24.jpg	Unknown	0.620401	1.35017	Falhou
JapaCN_25.jpg	Unknown	0.613788	1.33318	Falhou
JapaCN_27.jpg	JapaCN Nawan 7963226	0.455271	1.61032	Sucesso
JapaCN_28.jpg	Unknown	0.698155	1.32194	Falhou
JapaCN_30.jpg	Unknown	0.554231	1.36051	Falhou
JapaCN_31.jpg	Unknown	0.702884	1.30019	Falhou
JapaCN_32.jpg	Unknown	0.553436	1.33658	Falhou
JapaCN_34.jpg	JapaCN Nawan 7963226	0.46611	1.67445	Sucesso
JapaWhite_0.jpg	Unknown	0.50715	1.33049	Falhou
JapaWhite_2.jpg	JapaWhite Houriish 2963226	0.444216	1.62489	Sucesso
JapaWhite_3.jpg	JapaWhite Houriish 2963226	0.454792	1.59724	Sucesso
JapaWhite_5.jpg	JapaWhite Houriish 2963226	0.418874	1.59805	Sucesso
JapaWhite_6.jpg	Unknown	0.564625	1.34	Falhou
JapaWhite_8.jpg	JapaWhite Houriish 2963226	0.395757	1.65084	Sucesso
JapaWhite_9.jpg	JapaWhite Houriish 2963226	0.392534	1.63156	Sucesso
JapaWhite_11.jpg	Unknown	0.513541	1.31573	Falhou
JapaWhite_12.jpg	Unknown	0.553938	1.32278	Falhou
JapaWhite_13.jpg	JapaWhite Houriish 2963226	0.424145	1.64205	Sucesso
JapaWhite_15.jpg	Unknown	0.553938	1.346	Falhou
JapaWhite_16.jpg	JapaWhite Houriish 2963226	0.449314	1.65565	Sucesso
JapaWhite_18.jpg	JapaWhite Houriish 2963226	0.304738	1.61102	Sucesso
JapaWhite_19.jpg	JapaWhite Houriish 2963226	0.270384	1.61779	Sucesso
JapaWhite_21.jpg	Unknown	0.5631049	1.28877	Falhou
JapaWhite_22.jpg	Unknown	0.50715	1.35089	Falhou
JapaWhite_24.jpg	JapaWhite Houriish 2963226	0.446333	1.64664	Sucesso
JapaWhite_25.jpg	Unknown	0.541965	1.40373	Falhou
JapaWhite_27.jpg	JapaWhite Houriish 2963226	0.304738	1.63924	Sucesso
JapaWhite_28.jpg	JapaWhite Houriish 2963226	0.446333	1.63606	Sucesso
JapaWhite_30.jpg	JapaWhite Houriish 2963226	0.454792	1.54686	Sucesso
JapaWhite_31.jpg	Unknown	0.541965	1.3715	Falhou
JapaWhite_33.jpg	JapaWhite Houriish 2963226	0.446333	1.65214	Sucesso
JapaWhite_34.jpg	JapaWhite Houriish 2963226	0.270384	1.66019	Sucesso
JapaOld_1.jpg	JapaOld Akira 1763226	0.486752	1.75016	Sucesso
JapaOld_3.jpg	Unknown	0.50658	1.39344	Falhou
JapaOld_4.jpg	JapaOld Akira 1763226	0.498652	1.64978	Sucesso
JapaOld_6.jpg	Unknown	0.583163	1.32041	Falhou
JapaOld_7.jpg	JapaOld Akira 1763226	0.498652	1.66621	Sucesso

JapaOld_8.jpg	Unknown	0.515169	1.45006	Falhou
JapaOld_10.jpg	Unknown	0.531213	1.41187	Falhou
JapaOld_11.jpg	Unknown	0.539633	1.44242	Falhou
JapaOld_13.jpg	Unknown	0.603359	1.37419	Falhou
JapaOld_14.jpg	Unknown	0.538284	1.38184	Falhou
JapaOld_15.jpg	Unknown	0.524804	1.37453	Falhou
JapaOld_17.jpg	Unknown	0.528713	1.36941	Falhou
JapaOld_18.jpg	Unknown	0.52448	1.40489	Falhou
JapaOld_19.jpg	Unknown	0.601655	1.42747	Falhou
JapaOld_21.jpg	Unknown	0.58411	1.34178	Falhou
JapaOld_22.jpg	Unknown	0.58411	1.32616	Falhou
JapaOld_23.jpg	Unknown	0.53101	1.36762	Falhou
JapaOld_25.jpg	Unknown	0.594037	1.39875	Falhou
JapaOld_26.jpg	Unknown	0.63417	1.4079	Falhou
JapaOld_27.jpg	Unknown	0.581423	1.38385	Falhou
JapaOld_29.jpg	Unknown	0.522792	1.39744	Falhou
JapaOld_30.jpg	Unknown	0.524804	1.38595	Falhou
JapaOld_31.jpg	JapaOld Akira 1763226	0.486752	1.69697	Sucesso
JapaOld_33.jpg	Unknown	0.53101	1.34491	Falhou
JapaOld_34.jpg	Unknown	0.538284	1.42612	Falhou
JapaShort_1.jpg	JapaShort Okuri 1232266	0.488683	1.66203	Sucesso
JapaShort_2.jpg	JapaShort Okuri 1232266	0.446024	1.66889	Sucesso
JapaShort_4.jpg	Unknown	0.509885	1.29408	Falhou
JapaShort_7.jpg	JapaShort Okuri 1232266	0.403095	1.63029	Sucesso
JapaShort_8.jpg	JapaShort Okuri 1232266	0.399425	1.67603	Sucesso
JapaShort_10.jpg	JapaShort Okuri 1232266	0.380453	1.68406	Sucesso
JapaShort_12.jpg	Unknown	0.570314	1.36056	Falhou
JapaShort_13.jpg	Unknown	0.622365	1.31983	Falhou
JapaShort_14.jpg	Unknown	0.640098	1.29676	Falhou
JapaShort_16.jpg	JapaShort Okuri 1232266	0.364138	1.68567	Sucesso
JapaShort_17.jpg	JapaShort Okuri 1232266	0.412823	1.67618	Sucesso
JapaShort_19.jpg	JapaShort Okuri 1232266	0.461353	1.61076	Sucesso
JapaShort_20.jpg	Unknown	0.642516	1.49023	Falhou
JapaShort_23.jpg	JapaShort Okuri 1232266	0.377721	1.64643	Sucesso
JapaShort_25.jpg	Unknown	0.516346	1.40151	Falhou
JapaShort_28.jpg	JapaShort Okuri 1232266	0.392135	1.68851	Sucesso
JapaShort_29.jpg	JapaShort Okuri 1232266	0.399425	1.71459	Sucesso
JapaShort_31.jpg	JapaShort Okuri 1232266	0.444716	1.78057	Sucesso
JapaHair_1.jpg	Unknown	0.507899	1.26982	Sucesso
JapaHair_3.jpg	Unknown	0.616936	1.24964	Sucesso
JapaHair_4.jpg	Unknown	0.525048	1.2879	Sucesso
JapaHair_5.jpg	JapaShort Okuri 1232266	0.404121	1.55202	Falhou
JapaHair_8.jpg	Unknown	0.598818	1.21589	Sucesso
JapaHair_11.jpg	JapaShort Okuri 1232266	0.433572	1.57735	Falhou
JapaHair_12.jpg	Unknown	0.562376	1.27591	Sucesso
JapaHair_14.jpg	Unknown	0.602046	1.27493	Sucesso
JapaHair_15.jpg	Unknown	0.517241	1.25962	Sucesso
JapaHair_16.jpg	Unknown	0.661137	1.40904	Sucesso
JapaHair_19.jpg	Unknown	0.642303	1.27924	Sucesso
JapaHair_22.jpg	JapaShort Okuri 1232266	0.445537	1.70599	Falhou
JapaHair_25.jpg	Unknown	0.527189	1.27691	Sucesso
JapaLong_0.jpg	Unknown	0.728169	1.23912	Falhou
JapaLong_1.jpg	Unknown	0.748857	1.26301	Falhou
JapaLong_2.jpg	Unknown	0.700863	1.28477	Falhou
JapaLong_3.jpg	Unknown	0.702915	1.36625	Falhou
JapaLong_6.jpg	Unknown	0.744095	1.30568	Falhou
JapaLong_7.jpg	Unknown	0.78871	1.29858	Falhou
JapaLong_9.jpg	Unknown	0.716258	1.26389	Falhou
JapaLong_10.jpg	Unknown	0.66456	1.26605	Falhou
JapaLong_11.jpg	Unknown	0.709995	1.31445	Falhou
JapaLong_13.jpg	Unknown	0.656752	1.31865	Falhou
JapaLong_15.jpg	Unknown	0.549622	1.29713	Falhou
JapaLong_16.jpg	Unknown	0.520298	1.30172	Falhou
JapaLong_19.jpg	Unknown	0.566434	1.28727	Falhou
JapaLong_21.jpg	Unknown	0.692666	1.32293	Falhou
JapaLong_23.jpg	Unknown	0.729627	1.26812	Falhou
JapaLong_24.jpg	Unknown	0.775852	1.28289	Falhou
JapaLong_25.jpg	Unknown	0.775852	1.26539	Falhou
JapaLong_26.jpg	Unknown	0.646453	1.26988	Falhou
JapaLong_28.jpg	Unknown	0.744095	1.29852	Falhou
JapaLong_29.jpg	Unknown	0.549622	1.31187	Falhou
JapaLong_30.jpg	Unknown	0.549622	1.30721	Falhou
JapaLong_31.jpg	Unknown	0.692666	1.31141	Falhou

Média	0.55	1.44
Desvio Padrão	0.11	0.23
Total Amostras	340	

Falhou	61.8%
Sucesso	37.9%

Tabela 2: Threshold = 0.6

Arquivo de Entrada	Nome Identificado	Similaridade	Tempo(s)	Resultado
Robson_0.jpg	Robson Silva 1111114	0.376122	1.63166	Sucesso
Robson_2.jpg	Robson Silva 1111114	0.440305	2.25735	Sucesso
Robson_4.jpg	Robson Silva 1111114	0.370024	1.55832	Sucesso
Robson_5.jpg	Robson Silva 1111114	0.415198	1.62645	Sucesso
Robson_7.jpg	Robson Silva 1111114	0.377051	1.61474	Sucesso
Robson_10.jpg	Robson Silva 1111114	0.521566	1.53821	Sucesso
Robson_12.jpg	Robson Silva 1111114	0.488629	1.54134	Sucesso
Robson_13.jpg	Robson Silva 1111114	0.360554	1.57156	Sucesso
Robson_15.jpg	Robson Silva 1111114	0.344918	1.57432	Sucesso
Robson_16.jpg	Robson Silva 1111114	0.401532	1.59281	Sucesso
Robson_20.jpg	Robson Silva 1111114	0.340411	1.56793	Sucesso
Robson_22.jpg	Robson Silva 1111114	0.509791	1.52754	Sucesso
Robson_23.jpg	Robson Silva 1111114	0.490274	1.57715	Sucesso
Robson_25.jpg	Robson Silva 1111114	0.444605	1.6121	Sucesso
Robson_26.jpg	Robson Silva 1111114	0.306372	1.57652	Sucesso
Robson_28.jpg	Robson Silva 1111114	0.414766	1.58938	Sucesso
Robson_29.jpg	Robson Silva 1111114	0.474255	1.53138	Sucesso
Robson_31.jpg	Robson Silva 1111114	0.346875	1.54595	Sucesso
Robson_32.jpg	Robson Silva 1111114	0.451032	1.57146	Sucesso
Robson_34.jpg	Robson Silva 1111114	0.509791	1.56563	Sucesso
Nune_3.jpg	Unknown	0.793143	1.26929	Falhou
Nune_5.jpg	Unknown	0.704727	1.23275	Falhou
Nune_6.jpg	Unknown	0.711003	1.8257	Falhou
Nune_8.jpg	Unknown	0.674126	1.18184	Falhou
Nune_11.jpg	Nune nascimento 1111112	0.599007	2.21746	Sucesso
Nune_16.jpg	Nune nascimento 1111112	0.529269	1.54133	Sucesso
Nune_19.jpg	Unknown	0.793143	1.26789	Falhou
Nune_22.jpg	Nune nascimento 1111112	0.52475	1.60547	Sucesso
Nune_25.jpg	Unknown	0.760392	1.23621	Falhou
Nune_26.jpg	Unknown	0.763784	1.21241	Falhou
Nune_27.jpg	Unknown	0.704464	2.44503	Falhou
Nune_33.jpg	Unknown	0.704727	1.68042	Falhou
Lucas_0.jpg	Lucas Telles 1111111	0.414195	1.60785	Sucesso
Lucas_1.jpg	Lucas Telles 1111111	0.57731	1.57882	Sucesso
Lucas_3.jpg	Lucas Telles 1111111	0.584095	1.63708	Sucesso
Lucas_5.jpg	Lucas Telles 1111111	0.584798	1.63038	Sucesso
Lucas_6.jpg	Unknown	0.630378	1.30598	Falhou
Lucas_7.jpg	Lucas Telles 1111111	0.595518	1.60148	Sucesso
Lucas_9.jpg	Unknown	0.633048	1.99219	Falhou
Lucas_11.jpg	Unknown	0.633015	1.25129	Falhou
Lucas_12.jpg	Lucas Telles 1111111	0.489177	1.64298	Sucesso
Lucas_14.jpg	Lucas Telles 1111111	0.588417	1.54794	Sucesso
Lucas_15.jpg	Lucas Telles 1111111	0.504723	1.55972	Sucesso
Lucas_17.jpg	Lucas Telles 1111111	0.393599	1.59698	Sucesso
Lucas_18.jpg	Unknown	0.625037	1.93977	Falhou
Lucas_20.jpg	Lucas Telles 1111111	0.435443	1.59443	Sucesso
Lucas_22.jpg	Lucas Telles 1111111	0.519442	1.54485	Sucesso
Lucas_23.jpg	Lucas Telles 1111111	0.438079	1.6248	Sucesso
Lucas_25.jpg	Lucas Telles 1111111	0.433368	2.23205	Sucesso
Lucas_27.jpg	Lucas Telles 1111111	0.575391	1.56028	Sucesso
Lucas_29.jpg	Unknown	0.656955	1.21138	Falhou
Lucas_30.jpg	Unknown	0.696459	1.24844	Falhou
Lucas_31.jpg	Lucas Telles 1111111	0.433651	2.31823	Sucesso
Rafael_1.jpg	Rafael Ribeiro 1111113	0.561113	2.33213	Sucesso
Rafael_3.jpg	Unknown	0.637384	1.23886	Falhou
Rafael_4.jpg	Rafael Ribeiro 1111113	0.504188	1.59159	Sucesso
Rafael_6.jpg	Rafael Ribeiro 1111113	0.556387	2.3031	Sucesso
Rafael_8.jpg	Rafael Ribeiro 1111113	0.551111	1.553	Sucesso
Rafael_9.jpg	Unknown	0.64709	1.28624	Falhou
Rafael_11.jpg	Rafael Ribeiro 1111113	0.555905	2.89425	Sucesso
Rafael_14.jpg	Rafael Ribeiro 1111113	0.526931	1.71751	Sucesso
Rafael_15.jpg	Rafael Ribeiro 1111113	0.473032	1.7949	Sucesso
Rafael_17.jpg	Rafael Ribeiro 1111113	0.537088	2.24083	Sucesso
Rafael_19.jpg	Unknown	0.634625	1.43356	Falhou
Rafael_20.jpg	Unknown	0.658907	1.2899	Falhou
Rafael_22.jpg	Unknown	0.60166	1.89409	Falhou
Rafael_24.jpg	Rafael Ribeiro 1111113	0.569201	1.57738	Sucesso
Rafael_25.jpg	Unknown	0.628585	1.29961	Falhou
Rafael_26.jpg	Unknown	0.660519	1.85484	Falhou
Rafael_28.jpg	Unknown	0.660336	2.53276	Falhou
Rafael_30.jpg	Rafael Ribeiro 1111113	0.574273	2.41276	Sucesso
Rafael_32.jpg	Unknown	0.64709	1.87011	Falhou
Rafael_34.jpg	Unknown	0.637384	1.2445	Falhou
Japa_0.jpg	Japa Kim 1622211	0.340739	2.51811	Sucesso
Japa_3.jpg	Japa Kim 1622211	0.309515	2.5657	Sucesso
Japa_5.jpg	Japa Kim 1622211	0.438068	2.39153	Sucesso
Japa_7.jpg	Japa Kim 1622211	0.368878	1.62006	Sucesso
Japa_9.jpg	Japa Kim 1622211	0.38349	1.65153	Sucesso
Japa_10.jpg	Japa Kim 1622211	0.385931	1.70375	Sucesso
Japa_14.jpg	Japa Kim 1622211	0.430921	2.30449	Sucesso
Japa_16.jpg	Japa Kim 1622211	0.509342	1.65038	Sucesso
Japa_17.jpg	Japa Kim 1622211	0.32825	2.39743	Sucesso
Japa_19.jpg	Japa Kim 1622211	0.430483	1.62399	Sucesso
Japa_22.jpg	Japa Kim 1622211	0.430822	2.15853	Sucesso
Japa_24.jpg	Japa Kim 1622211	0.461595	2.29927	Sucesso
Japa_26.jpg	Japa Kim 1622211	0.367067	1.66128	Sucesso
Japa_28.jpg	Japa Kim 1622211	0.423679	1.62725	Sucesso
Japa_30.jpg	Japa Kim 1622211	0.403724	2.39743	Sucesso
Japa_32.jpg	Japa Kim 1622211	0.403575	2.40665	Sucesso
JapaMan_1.jpg	JapaMan KimK 1234567	0.595951	2.36481	Sucesso
JapaMan_5.jpg	JapaMan KimK 1234567	0.568504	1.64756	Sucesso
JapaMan_9.jpg	JapaMan KimK 1234567	0.472461	2.36021	Sucesso
JapaMan_12.jpg	JapaMan KimK 1234567	0.450699	2.37815	Sucesso
JapaMan_14.jpg	JapaMan KimK 1234567	0.505556	2.35963	Sucesso
JapaMan_18.jpg	JapaMan KimK 1234567	0.49724	1.76768	Sucesso
JapaMan_19.jpg	JapaMan KimK 1234567	0.479664	1.62912	Sucesso
JapaMan_21.jpg	JapaMan KimK 1234567	0.422488	1.64744	Sucesso
JapaMan_23.jpg	JapaMan KimK 1234567	0.571361	1.59468	Sucesso
JapaMan_24.jpg	JapaMan KimK 1234567	0.584103	2.906	Sucesso
JapaMan_29.jpg	JapaMan KimK 1234567	0.434617	1.71229	Sucesso
JapaMan_31.jpg	JapaMan KimK 1234567	0.322308	1.65414	Sucesso
JapaChild_0.jpg	Unknown	0.606864	2.11277	Falhou

JapaChild_2.jpg	JapaChild Assaki 14132266	0.53199	2.28217	Sucesso
JapaChild_4.jpg	JapaChild Assaki 14132266	0.370904	1.60942	Sucesso
JapaChild_5.jpg	JapaChild Assaki 14132266	0.578729	1.67168	Sucesso
JapaChild_7.jpg	Unknown	0.612269	1.96423	Falhou
JapaChild_9.jpg	JapaChild Assaki 14132266	0.578729	1.74654	Sucesso
JapaChild_11.jpg	JapaChild Assaki 14132266	0.536591	1.61888	Sucesso
JapaChild_16.jpg	JapaChild Assaki 14132266	0.573721	1.61221	Sucesso
JapaChild_17.jpg	JapaChild Assaki 14132266	0.517901	1.59362	Sucesso
JapaChild_24.jpg	Unknown	0.606864	1.84742	Falhou
JapaChild_26.jpg	JapaChild Assaki 14132266	0.509596	2.32728	Sucesso
JapaChild_28.jpg	Unknown	0.612269	2.02766	Falhou
JapaChild_31.jpg	JapaChild Assaki 14132266	0.569156	1.68182	Sucesso
JapaChild_33.jpg	Unknown	0.647537	1.34994	Falhou
JapaChild_34.jpg	Unknown	0.606864	2.07025	Falhou
JapaNew_1.jpg	Unknown	0.605386	2.16628	Falhou
JapaNew_4.jpg	Unknown	0.637109	2.06781	Falhou
JapaNew_7.jpg	JapaNew Assaki 123226	0.575926	1.66446	Sucesso
JapaNew_9.jpg	JapaNew Assaki 123226	0.576967	1.64658	Sucesso
JapaNew_10.jpg	Unknown	0.624824	1.94396	Falhou
JapaNew_12.jpg	Unknown	0.631206	1.29381	Falhou
JapaNew_13.jpg	Unknown	0.647633	1.29513	Falhou
JapaNew_15.jpg	JapaNew Assaki 123226	0.540276	2.30329	Sucesso
JapaNew_17.jpg	JapaNew Assaki 123226	0.587175	2.89492	Sucesso
JapaNew_19.jpg	JapaNew Assaki 123226	0.577369	2.47817	Sucesso
JapaNew_22.jpg	Unknown	0.632874	1.9283	Falhou
JapaNew_24.jpg	Unknown	0.625342	1.27443	Falhou
JapaNew_27.jpg	Unknown	0.644837	2.21525	Falhou
JapaNew_29.jpg	Unknown	0.641514	1.94641	Falhou
JapaNew_31.jpg	Unknown	0.616136	1.98532	Falhou
JapaNew_33.jpg	Unknown	0.605386	1.96968	Falhou
JapaFat_0.jpg	Unknown	0.600604	1.35071	Sucesso
JapaFat_1.jpg	Unknown	0.600604	1.35073	Sucesso
JapaFat_2.jpg	Unknown	0.600604	2.10278	Sucesso
JapaFat_6.jpg	Unknown	0.617925	1.35904	Sucesso
JapaFat_8.jpg	Unknown	0.617925	1.34978	Sucesso
JapaFat_9.jpg	JapaChild Assaki 14132266	0.57749	2.43843	Falhou
JapaFat_11.jpg	JapaChild Assaki 14132266	0.594339	1.66192	Falhou
JapaFat_13.jpg	JapaChild Assaki 14132266	0.540146	1.6309	Falhou
JapaFat_15.jpg	JapaChild Assaki 14132266	0.504349	2.39948	Falhou
JapaFat_17.jpg	Unknown	0.637688	1.29139	Sucesso
JapaFat_18.jpg	JapaWhite Houriish 2963226	0.592161	1.68945	Falhou
JapaFat_20.jpg	JapaChild Assaki 14132266	0.594319	1.71188	Falhou
JapaFat_21.jpg	JapaNew Assaki 123226	0.516561	1.69082	Falhou
JapaFat_23.jpg	JapaChild Assaki 14132266	0.569124	1.63488	Falhou
JapaFat_25.jpg	JapaNew Assaki 123226	0.461099	1.57478	Falhou
JapaFat_26.jpg	JapaNew Assaki 123226	0.461099	1.59782	Falhou
JapaFat_29.jpg	Unknown	0.608981	1.37058	Sucesso
JapaFat_31.jpg	JapaNew Assaki 123226	0.533811	1.67666	Falhou
JapaFat_32.jpg	Unknown	0.601084	1.36135	Sucesso
JapaFat_34.jpg	Unknown	0.601084	1.32558	Sucesso
JapaUS_0.jpg	Unknown	0.720267	1.27092	Falhou
JapaUS_1.jpg	Unknown	0.735039	1.97108	Falhou
JapaUS_3.jpg	Unknown	0.696113	1.97063	Falhou
JapaUS_5.jpg	Unknown	0.71646	2.1688	Falhou
JapaUS_7.jpg	Unknown	0.625901	2.17716	Falhou
JapaUS_9.jpg	JapaUS Nashira 4963226	0.581703	1.57693	Sucesso
JapaUS_10.jpg	Unknown	0.656526	1.30516	Falhou
JapaUS_12.jpg	Unknown	0.68989	2.05736	Falhou
JapaUS_14.jpg	Unknown	0.714062	1.27769	Falhou
JapaUS_15.jpg	Unknown	0.717032	1.29301	Falhou
JapaUS_16.jpg	Unknown	0.725941	1.31974	Falhou
JapaUS_17.jpg	Unknown	0.712146	1.9165	Falhou
JapaUS_19.jpg	Unknown	0.70983	1.29394	Falhou
JapaUS_21.jpg	Unknown	0.702007	1.24525	Falhou
JapaUS_22.jpg	Unknown	0.686911	1.30816	Falhou
JapaUS_23.jpg	Unknown	0.651061	1.98535	Falhou
JapaUS_25.jpg	Unknown	0.644131	1.30375	Falhou
JapaUS_26.jpg	Unknown	0.687146	1.30766	Falhou
JapaUS_28.jpg	Unknown	0.702007	1.24722	Falhou
JapaUS_29.jpg	Unknown	0.720267	1.92359	Falhou
JapaUS_31.jpg	Unknown	0.717032	1.26758	Falhou
JapaUS_32.jpg	Unknown	0.656526	1.31795	Falhou
JapaUS_33.jpg	Unknown	0.642395	1.30974	Falhou
JapaUS_34.jpg	Unknown	0.686911	1.94517	Falhou
JapaCN_1.jpg	JapaCN Nawan 7963226	0.497959	1.63353	Sucesso
JapaCN_3.jpg	JapaCN Nawan 7963226	0.573734	1.74846	Sucesso
JapaCN_5.jpg	JapaCN Nawan 7963226	0.505968	1.58098	Sucesso
JapaCN_6.jpg	JapaCN Nawan 79632			

JapaWhite_15.jpg	JapaWhite Houriish 2963226	0.553938	1.6344	Sucesso
JapaWhite_16.jpg	JapaWhite Houriish 2963226	0.449314	1.61287	Sucesso
JapaWhite_18.jpg	JapaWhite Houriish 2963226	0.304738	1.5983	Sucesso
JapaWhite_20.jpg	JapaWhite Houriish 2963226	0.468884	1.59136	Sucesso
JapaWhite_21.jpg	JapaWhite Houriish 2963226	0.561049	1.63287	Sucesso
JapaWhite_23.jpg	JapaWhite Houriish 2963226	0.463538	1.65035	Sucesso
JapaWhite_24.jpg	JapaWhite Houriish 2963226	0.446333	2.84736	Sucesso
JapaWhite_27.jpg	JapaWhite Houriish 2963226	0.304738	2.39032	Sucesso
JapaWhite_29.jpg	JapaWhite Houriish 2963226	0.513541	2.3569	Sucesso
JapaWhite_32.jpg	JapaWhite Houriish 2963226	0.270384	1.6196	Sucesso
JapaWhite_33.jpg	JapaWhite Houriish 2963226	0.446333	2.35564	Sucesso
JapaOld_1.jpg	JapaOld Akira 1763226	0.486752	2.46531	Sucesso
JapaOld_3.jpg	JapaOld Akira 1763226	0.50658	1.68029	Sucesso
JapaOld_5.jpg	JapaOld Akira 1763226	0.448596	1.58861	Sucesso
JapaOld_6.jpg	JapaOld Akira 1763226	0.583163	1.66317	Sucesso
JapaOld_8.jpg	JapaOld Akira 1763226	0.515169	1.73654	Sucesso
JapaOld_9.jpg	JapaOld Akira 1763226	0.510509	1.72953	Sucesso
JapaOld_11.jpg	JapaOld Akira 1763226	0.539633	1.76072	Sucesso
JapaOld_13.jpg	Unknown	0.603359	1.374	Falhou
JapaOld_14.jpg	JapaOld Akira 1763226	0.538284	1.73748	Sucesso
JapaOld_16.jpg	JapaOld Akira 1763226	0.50658	1.71429	Sucesso
JapaOld_18.jpg	JapaOld Akira 1763226	0.52448	2.42294	Sucesso
JapaOld_20.jpg	Unknown	0.604374	1.4603	Falhou
JapaOld_21.jpg	JapaOld Akira 1763226	0.58411	1.62774	Sucesso
JapaOld_23.jpg	JapaOld Akira 1763226	0.53101	1.65132	Sucesso
JapaOld_25.jpg	JapaOld Akira 1763226	0.594037	1.69891	Sucesso
JapaOld_26.jpg	Unknown	0.63417	1.40576	Falhou
JapaOld_28.jpg	JapaOld Akira 1763226	0.536089	2.34989	Sucesso
JapaOld_30.jpg	JapaOld Akira 1763226	0.524804	1.66606	Sucesso
JapaOld_31.jpg	JapaOld Akira 1763226	0.486752	1.64001	Sucesso
JapaOld_33.jpg	JapaOld Akira 1763226	0.53101	1.66363	Sucesso
JapaShort_0.jpg	JapaShort Okuri 1232266	0.444716	1.63791	Sucesso
JapaShort_1.jpg	JapaShort Okuri 1232266	0.488683	1.67581	Sucesso
JapaShort_3.jpg	JapaShort Okuri 1232266	0.500444	1.65185	Sucesso
JapaShort_6.jpg	JapaShort Okuri 1232266	0.403095	2.38364	Sucesso
JapaShort_8.jpg	JapaShort Okuri 1232266	0.399425	2.51241	Sucesso
JapaShort_11.jpg	JapaShort Okuri 1232266	0.384463	1.61922	Sucesso
JapaShort_12.jpg	JapaShort Okuri 1232266	0.570314	2.35702	Sucesso
JapaShort_15.jpg	JapaShort Okuri 1232266	0.399556	2.33323	Sucesso
JapaShort_17.jpg	JapaShort Okuri 1232266	0.412823	2.32787	Sucesso

JapaShort_19.jpg	JapaShort Okuri 1232266	0.461353	2.40877	Sucesso
JapaShort_21.jpg	Unknown	0.652922	1.4615	Falhou
JapaShort_23.jpg	JapaShort Okuri 1232266	0.377721	2.27719	Sucesso
JapaShort_25.jpg	JapaShort Okuri 1232266	0.516346	2.47026	Sucesso
JapaShort_28.jpg	JapaShort Okuri 1232266	0.392135	1.68961	Sucesso
JapaShort_30.jpg	JapaShort Okuri 1232266	0.399556	1.68512	Sucesso
JapaHair_0.jpg	JapaWhite Houriish 2963226	0.53529	2.24432	Falhou
JapaHair_3.jpg	Unknown	0.616936	1.21286	Sucesso
JapaHair_4.jpg	JapaUS Nashira 4963226	0.525048	1.55221	Falhou
JapaHair_5.jpg	JapaShort Okuri 1232266	0.404121	2.21448	Falhou
JapaHair_8.jpg	JapaWhite Houriish 2963226	0.596818	1.52865	Falhou
JapaHair_11.jpg	JapaShort Okuri 1232266	0.433572	2.30047	Falhou
JapaHair_13.jpg	Unknown	0.647445	1.23773	Sucesso
JapaHair_14.jpg	Unknown	0.602046	1.27117	Sucesso
JapaHair_16.jpg	Unknown	0.661137	1.39828	Sucesso
JapaHair_19.jpg	Unknown	0.642303	1.28789	Sucesso
JapaHair_22.jpg	JapaShort Okuri 1232266	0.445537	2.32955	Falhou
JapaHair_25.jpg	JapaUS Nashira 4963226	0.527189	1.56533	Falhou
JapaLong_0.jpg	Unknown	0.728169	2.03341	Falhou
JapaLong_2.jpg	Unknown	0.700863	1.93729	Falhou
JapaLong_7.jpg	Unknown	0.78871	1.27853	Falhou
JapaLong_8.jpg	Unknown	0.763297	1.2877	Falhou
JapaLong_9.jpg	Unknown	0.716258	1.26164	Falhou
JapaLong_10.jpg	Unknown	0.66456	1.75544	Falhou
JapaLong_13.jpg	Unknown	0.656752	1.96596	Falhou
JapaLong_15.jpg	JapaLong Oishi 1232257	0.549622	2.44606	Sucesso
JapaLong_19.jpg	JapaLong Oishi 1232257	0.566434	1.60652	Sucesso
JapaLong_22.jpg	Unknown	0.735833	1.25126	Falhou
JapaLong_23.jpg	Unknown	0.729627	1.25078	Falhou
JapaLong_25.jpg	Unknown	0.775852	1.25403	Falhou
JapaLong_26.jpg	Unknown	0.646453	1.91488	Falhou
JapaLong_28.jpg	Unknown	0.744095	1.28897	Falhou
JapaLong_29.jpg	JapaLong Oishi 1232257	0.549622	1.62987	Sucesso
JapaLong_31.jpg	Unknown	0.692666	1.27338	Falhou
Média		0.55	1.76	
Desvio Padrão		0.11	0.40	
Total Amostras		279		
		Falhou	39.6%	
		Sucesso	60.4%	

Tabela 3: Threshold = 0.7

Arquivo de Entrada	Nome Identificado	Similaridade	Tempo(s)	Resultado
Robson_1.jpg	Robson Silva 1111114	0.349555	1.52616	Sucesso
Robson_2.jpg	Robson Silva 1111114	0.440305	1.53852	Sucesso
Robson_4.jpg	Robson Silva 1111114	0.370024	1.56265	Sucesso
Robson_5.jpg	Robson Silva 1111114	0.415198	1.6142	Sucesso
Robson_7.jpg	Robson Silva 1111114	0.377051	1.59602	Sucesso
Robson_10.jpg	Robson Silva 1111114	0.521566	1.51878	Sucesso
Robson_12.jpg	Robson Silva 1111114	0.488629	1.52198	Sucesso
Robson_13.jpg	Robson Silva 1111114	0.360554	1.57241	Sucesso
Robson_15.jpg	Robson Silva 1111114	0.344918	1.55181	Sucesso
Robson_16.jpg	Robson Silva 1111114	0.401532	1.58813	Sucesso
Robson_19.jpg	Robson Silva 1111114	0.624439	1.52964	Sucesso
Robson_21.jpg	Robson Silva 1111114	0.572722	1.5307	Sucesso
Robson_22.jpg	Robson Silva 1111114	0.509791	1.53359	Sucesso
Robson_24.jpg	Robson Silva 1111114	0.317554	1.60649	Sucesso
Robson_25.jpg	Robson Silva 1111114	0.444605	1.59719	Sucesso
Robson_27.jpg	Robson Silva 1111114	0.392245	1.61701	Sucesso
Robson_28.jpg	Robson Silva 1111114	0.414766	1.59387	Sucesso
Robson_31.jpg	Robson Silva 1111114	0.346875	1.51875	Sucesso
Robson_32.jpg	Robson Silva 1111114	0.451032	1.58865	Sucesso
Robson_34.jpg	Robson Silva 1111114	0.509791	1.56333	Sucesso
Nune_3.jpg	Unknown	0.793143	1.25608	Falhou
Nune_5.jpg	Unknown	0.704727	1.23782	Falhou
Nune_6.jpg	Unknown	0.711003	1.17307	Falhou
Nune_11.jpg	Nune nascimento 1111112	0.599007	1.49242	Sucesso
Nune_16.jpg	Nune nascimento 1111112	0.529269	1.48805	Sucesso
Nune_19.jpg	Unknown	0.793143	1.24226	Falhou
Nune_20.jpg	Unknown	0.789155	1.92475	Falhou
Nune_22.jpg	Nune nascimento 1111112	0.52475	1.58486	Sucesso
Nune_25.jpg	Unknown	0.760392	1.25619	Falhou
Nune_26.jpg	Unknown	0.763784	1.23763	Falhou
Nune_33.jpg	Unknown	0.704727	1.24448	Falhou
Nune_34.jpg	Unknown	0.793143	2.02358	Falhou
Lucas_1.jpg	Lucas Telles 1111111	0.57731	1.57797	Sucesso
Lucas_2.jpg	Lucas Telles 1111111	0.544156	2.39361	Sucesso
Lucas_5.jpg	Lucas Telles 1111111	0.584798	2.08437	Sucesso
Lucas_7.jpg	Lucas Telles 1111111	0.595518	1.57857	Sucesso
Lucas_8.jpg	Lucas Telles 1111111	0.670188	1.57281	Sucesso
Lucas_10.jpg	Lucas Telles 1111111	0.61336	2.24686	Sucesso
Lucas_12.jpg	Lucas Telles 1111111	0.489177	1.59345	Sucesso
Lucas_13.jpg	Lucas Telles 1111111	0.492203	1.65007	Sucesso
Lucas_15.jpg	Lucas Telles 1111111	0.504723	1.56215	Sucesso
Lucas_17.jpg	Lucas Telles 1111111	0.393599	1.57201	Sucesso
Lucas_18.jpg	Lucas Telles 1111111	0.625307	1.56979	Sucesso
Lucas_20.jpg	Lucas Telles 1111111	0.435443	1.56094	Sucesso
Lucas_21.jpg	Lucas Telles 1111111	0.491181	2.33568	Sucesso
Lucas_23.jpg	Lucas Telles 1111111	0.438079	1.61349	Sucesso
Lucas_25.jpg	Lucas Telles 1111111	0.433368	1.55152	Sucesso
Lucas_26.jpg	Lucas Telles 1111111	0.408063	2.30061	Sucesso
Lucas_29.jpg	Lucas Telles 1111111	0.656955	1.51129	Sucesso
Lucas_30.jpg	Lucas Telles 1111111	0.699459	1.56741	Sucesso
Lucas_32.jpg	Lucas Telles 1111111	0.653958	2.32699	Sucesso
Lucas_34.jpg	Lucas Telles 1111111	0.432173	1.58722	Sucesso
Rafael_1.jpg	Rafael Ribeiro 1111113	0.561113	1.59105	Sucesso
Rafael_3.jpg	Rafael Ribeiro 1111113	0.637384	1.54666	Sucesso
Rafael_5.jpg	Rafael Ribeiro 1111113	0.466501	2.34596	Sucesso
Rafael_7.jpg	Rafael Ribeiro 1111113	0.599975	1.55714	Sucesso
Rafael_8.jpg	Rafael Ribeiro 1111113	0.551111	1.55694	Sucesso
Rafael_10.jpg	Rafael Ribeiro 1111113	0.547529	1.58745	Sucesso
Rafael_11.jpg	Rafael Ribeiro 1111113	0.555905	1.56728	Sucesso
Rafael_13.jpg	Rafael Ribeiro 1111113	0.577498	2.21026	Sucesso
Rafael_15.jpg	Rafael Ribeiro 1111113	0.473032	1.79591	Sucesso
Rafael_17.jpg	Rafael Ribeiro 1111113	0.537088	1.56473	Sucesso
Rafael_18.jpg	Rafael Ribeiro 1111113	0.548976	2.19823	Sucesso
Rafael_20.jpg	Rafael Ribeiro 1111113	0.658907	1.58502	Sucesso
Rafael_22.jpg	Rafael Ribeiro 1111113	0.60166	1.51743	Sucesso
Rafael_24.jpg	Rafael Ribeiro 1111113	0.569201	1.54762	Sucesso
Rafael_25.jpg	Rafael Ribeiro 1111113	0.628585	1.60491	Sucesso
Rafael_27.jpg	Rafael Ribeiro 1111113	0.584062	1.57049	Sucesso
Rafael_28.jpg	Rafael Ribeiro 1111113	0.660336	1.59864	Sucesso
Rafael_30.jpg	Rafael Ribeiro 1111113	0.574273	1.73746	Sucesso
Rafael_31.jpg	Rafael Ribeiro 1111113	0.658907	2.26781	Sucesso
Rafael_34.jpg	Rafael Ribeiro 1111113	0.637384	1.53613	Sucesso
Japa_0.jpg	Japa Kim 1622211	0.340739	1.71089	Sucesso
Japa_2.jpg	Japa Kim 1622211	0.423085	1.58996	Sucesso
Japa_3.jpg	Japa Kim 1622211	0.309515	1.67322	Sucesso
Japa_5.jpg	Japa Kim 1622211	0.438068	1.68671	Sucesso
Japa_7.jpg	Japa Kim 1622211	0.368878	1.58727	Sucesso
Japa_8.jpg	Japa Kim 1622211	0.446099	1.88626	Sucesso
Japa_10.jpg	Japa Kim 1622211	0.385931	1.69648	Sucesso
Japa_15.jpg	Japa Kim 1622211	0.445857	1.67229	Sucesso
Japa_17.jpg	Japa Kim 1622211	0.483096	1.6207	Sucesso
Japa_18.jpg	Japa Kim 1622211	0.406627	1.65877	Sucesso
Japa_20.jpg	Japa Kim 1622211	0.475414	1.65639	Sucesso
Japa_22.jpg	Japa Kim 1622211	0.430822	1.67456	Sucesso
Japa_23.jpg	Japa Kim 1622211	0.503496	1.67661	Sucesso
Japa_25.jpg	Japa Kim 1622211	0.413249	2.51374	Sucesso
Japa_27.jpg	Japa Kim 1622211	0.412012	2.5211	Sucesso
Japa_30.jpg	Japa Kim 1622211	0.32825	1.67751	Sucesso
Japa_31.jpg	Japa Kim 1622211	0.376852	2.36071	Sucesso
JapaMan_1.jpg	JapaMan KimK 1234567	0.595951	1.6064	Sucesso
JapaMan_5.jpg	JapaMan KimK 1234567	0.568504	1.63365	Sucesso
JapaMan_8.jpg	JapaMan KimK 1234567	0.554114	1.67975	Sucesso
JapaMan_10.jpg	JapaMan KimK 1234567	0.464239	2.94136	Sucesso
JapaMan_12.jpg	JapaMan KimK 1234567	0.450699	1.6315	Sucesso
JapaMan_14.jpg	JapaMan KimK 1234567	0.505556	1.60297	Sucesso
JapaMan_19.jpg	JapaMan KimK 1234567	0.479664	1.71367	Sucesso
JapaMan_21.jpg	JapaMan KimK 1234567	0.422488	1.63404	Sucesso

JapaMan_23.jpg	JapaMan KimK 1234567	0.571361	1.5808	Sucesso
JapaMan_24.jpg	JapaMan KimK 1234567	0.584103	1.60031	Sucesso
JapaMan_26.jpg	JapaMan KimK 1234567	0.655137	1.60146	Sucesso
JapaMan_29.jpg	JapaMan KimK 1234567	0.434617	1.64974	Sucesso
JapaMan_30.jpg	JapaMan KimK 1234567	0.460389	2.32142	Sucesso
JapaChild_0.jpg	JapaChild Assaki 14132266	0.606864	1.56025	Sucesso
JapaChild_1.jpg	JapaChild Assaki 14132266	0.533522	1.62218	Sucesso
JapaChild_3.jpg	JapaChild Assaki 14132266	0.606864	2.28129	Sucesso
JapaChild_5.jpg	JapaChild Assaki 14132266	0.578729	1.65829	Sucesso
JapaChild_7.jpg	JapaChild Assaki 14132266	0.612269	1.61593	Sucesso
JapaChild_8.jpg	JapaChild Assaki 14132266	0.643818	2.7392	Sucesso
JapaChild_11.jpg	JapaChild Assaki 14132266	0.536591	1.61139	Sucesso
JapaChild_15.jpg	JapaChild Assaki 14132266	0.63906	1.64948	Sucesso
JapaChild_17.jpg	JapaChild Assaki 14132266	0.517901	1.5524	Sucesso
JapaChild_24.jpg	JapaChild Assaki 14132266	0.606864	1.60486	Sucesso
JapaChild_25.jpg	JapaChild Assaki 14132266	0.533522	1.63749	Sucesso
JapaChild_28.jpg	JapaChild Assaki 14132266	0.612269	1.64419	Sucesso
JapaChild_31.jpg	JapaChild Assaki 14132266	0.569156	1.64859	Sucesso
JapaChild_32.jpg	JapaChild Assaki 14132266	0.62749	1.62763	Sucesso
JapaChild_34.jpg	JapaChild Assaki 14132266	0.606864	1.61653	Sucesso
JapaNew_0.jpg	JapaNew Assaki 1223226	0.677201	1.64131	Sucesso
JapaNew_2.jpg	JapaNew Assaki 1223226	0.650356	1.58719	Sucesso
JapaNew_3.jpg	JapaNew Assaki 1223226	0.637109	2.44594	Sucesso
JapaNew_7.jpg	JapaNew Assaki 1223226	0.575926	1.63752	Sucesso
JapaNew_8.jpg	JapaNew Assaki 1223226	0.575926	2.5287	Sucesso
JapaNew_11.jpg	JapaNew Assaki 1223226	0.677201	1.60196	Sucesso
JapaNew_12.jpg	JapaNew Assaki 1223226	0.631206	1.59487	Sucesso
JapaNew_16.jpg	JapaNew Assaki 1223226	0.616136	2.42711	Sucesso
JapaNew_18.jpg	JapaNew Assaki 1223226	0.587175	1.65424	Sucesso
JapaNew_20.jpg	JapaNew Assaki 1223226	0.577369	1.6044	Sucesso
JapaNew_21.jpg	JapaNew Assaki 1223226	0.633068	1.61356	Sucesso
JapaNew_23.jpg	JapaNew Assaki 1223226	0.640888	1.55935	Sucesso
JapaNew_24.jpg	JapaNew Assaki 1223226	0.625342	1.61722	Sucesso
JapaNew_26.jpg	JapaNew Assaki 1223226	0.591778	1.64958	Sucesso
JapaNew_27.jpg	JapaNew Assaki 1223226	0.644837	1.63617	Sucesso
JapaNew_29.jpg	JapaNew Assaki 1223226	0.641514	1.58257	Sucesso
JapaNew_31.jpg	JapaNew Assaki 1223226	0.616136	1.62704	Sucesso
JapaNew_32.jpg	JapaNew Assaki 1223226	0.650356	1.59372	Sucesso
JapaNew_34.jpg	JapaNew Assaki 1223226	0.616136	1.67209	Sucesso
JapaFat_0.jpg	JapaWhite Houriish 2963226	0.600604	1.66388	Falhou
JapaFat_2.jpg	JapaWhite Houriish 2963226	0.600604	1.66365	Falhou
JapaFat_6.jpg	JapaWhite Houriish 2963226	0.617925	1.65196	Falhou
JapaFat_8.jpg	JapaWhite Houriish 2963226	0.617925	1.64385	Falhou
JapaFat_9.jpg	JapaChild Assaki 14132266	0.57749	1.66258	Falhou
JapaFat_11.jpg	JapaChild Assaki 14132266	0.594339	1.65091	Falhou
JapaFat_13.jpg	JapaChild Assaki 14132266	0.540146	1.60451	Falhou
JapaFat_14.jpg	JapaChild Assaki 14132266	0.504349	1.60847	Falhou
JapaFat_16.jpg	JapaChild Assaki 14132266	0.504349	1.58404	Falhou
JapaFat_17.jpg	JapaWhite Houriish 2963226	0.637688	1.5999	Falhou
JapaFat_19.jpg	JapaChild Assaki 14132266	0.594319	2.46998	Falhou
JapaFat_21.jpg	JapaNew Assaki 1223226	0.51651	1.67881	Falhou
JapaFat_23.jpg	JapaChild Assaki 14132266	0.569124	1.63251	Falhou
JapaFat_25.jpg	JapaNew Assaki 1223226	0.461099	1.56341	Falhou
JapaFat_26.jpg	JapaNew Assaki 1223226	0.461099	1.5806	Falhou
JapaFat_28.jpg	JapaNew Assaki 1223226	0.548177	1.55792	Falhou
JapaFat_29.jpg	JapaWhite Houriish 2963226	0.608981	1.67066	Falhou
JapaFat_31.jpg	JapaNew Assaki 1223226	0.533811	1.66291	Falhou
JapaFat_32.jpg	JapaWhite Houriish 2963226	0.601084	1.67343	Falhou
JapaFat_34.jpg	JapaWhite Houriish 2963226	0.601084	1.63599	Falhou
JapaUS_1.jpg	Unknown	0.735039	1.26294	Falhou
JapaUS_2.jpg	JapaUS Nashira 4963226	0.696113	1.55026	Sucesso
JapaUS_3.jpg	JapaUS Nashira 4963226	0.696113	2.30265	Sucesso
JapaUS_5.jpg	Unknown	0.71646	1.29315	Falhou
JapaUS_6.jpg	JapaUS Nashira 4963226	0.642395	2.38631	Sucesso
JapaUS_9.jpg	JapaUS Nashira 4963226	0.581703	1.56117	Sucesso
JapaUS_10.jpg	JapaUS Nashira 4963226	0.656526	1.59386	Sucesso
JapaUS_12.jpg	JapaUS Nashira 4963226	0.688989	1.57697	Sucesso
JapaUS_13.jpg	Unknown	0.706908	1.95632	Falhou
JapaUS_15.jpg	Unknown	0.717032	1.29669	Falhou
JapaUS_17.jpg	Unknown	0.712146	1.28939	Falhou
JapaUS_18.jpg	JapaUS Nashira 4963226	0.686525	1.6004	Sucesso
JapaUS_19.jpg	Unknown	0.709		

JapaCN_25.jpg	JapaCN Nawan 7963226	0.63108	1.65303	Sucesso
JapaCN_27.jpg	JapaCN Nawan 7963226	0.455271	2.39663	Sucesso
JapaCN_29.jpg	Unknown	0.703586	1.28021	Falhou
JapaCN_30.jpg	JapaCN Nawan 7963226	0.554231	1.66311	Sucesso
JapaCN_32.jpg	JapaCN Nawan 7963226	0.553436	1.58407	Sucesso
JapaCN_33.jpg	JapaCN Nawan 7963226	0.620401	1.63937	Sucesso
JapaWhite_0.jpg	JapaWhite Hourish 2963226	0.50715	1.63134	Sucesso
JapaWhite_2.jpg	JapaWhite Hourish 2963226	0.444216	1.6292	Sucesso
JapaWhite_3.jpg	JapaWhite Hourish 2963226	0.454792	2.25116	Sucesso
JapaWhite_5.jpg	JapaWhite Hourish 2963226	0.418874	1.58398	Sucesso
JapaWhite_7.jpg	JapaWhite Hourish 2963226	0.46727	1.59531	Sucesso
JapaWhite_8.jpg	JapaWhite Hourish 2963226	0.395757	2.19857	Sucesso
JapaWhite_11.jpg	JapaWhite Hourish 2963226	0.513541	1.64304	Sucesso
JapaWhite_12.jpg	JapaWhite Hourish 2963226	0.553938	1.65108	Sucesso
JapaWhite_14.jpg	JapaWhite Hourish 2963226	0.50715	1.65703	Sucesso
JapaWhite_15.jpg	JapaWhite Hourish 2963226	0.553938	1.63904	Sucesso
JapaWhite_17.jpg	JapaWhite Hourish 2963226	0.424145	1.59986	Sucesso
JapaWhite_19.jpg	JapaWhite Hourish 2963226	0.270384	1.5902	Sucesso
JapaWhite_20.jpg	JapaWhite Hourish 2963226	0.46884	1.5996	Sucesso
JapaWhite_22.jpg	JapaWhite Hourish 2963226	0.50715	2.48899	Sucesso
JapaWhite_25.jpg	JapaWhite Hourish 2963226	0.541965	2.38974	Sucesso
JapaWhite_27.jpg	JapaWhite Hourish 2963226	0.304738	1.57074	Sucesso
JapaWhite_28.jpg	JapaWhite Hourish 2963226	0.446333	2.302	Sucesso
JapaWhite_31.jpg	JapaWhite Hourish 2963226	0.541965	2.33735	Sucesso
JapaWhite_33.jpg	JapaWhite Hourish 2963226	0.446333	1.62194	Sucesso
JapaWhite_34.jpg	JapaWhite Hourish 2963226	0.270384	1.64267	Sucesso
JapaOld_1.jpg	JapaOld Akira 1763226	0.486752	1.65645	Sucesso
JapaOld_2.jpg	JapaOld Akira 1763226	0.583163	1.66119	Sucesso
JapaOld_4.jpg	JapaOld Akira 1763226	0.498652	1.68227	Sucesso
JapaOld_6.jpg	JapaOld Akira 1763226	0.583163	1.64356	Sucesso
JapaOld_7.jpg	JapaOld Akira 1763226	0.498652	1.67428	Sucesso
JapaOld_9.jpg	JapaOld Akira 1763226	0.510509	1.71523	Sucesso
JapaOld_11.jpg	JapaOld Akira 1763226	0.539633	1.7148	Sucesso
JapaOld_12.jpg	JapaOld Akira 1763226	0.551928	1.69769	Sucesso
JapaOld_14.jpg	JapaOld Akira 1763226	0.538284	1.70926	Sucesso
JapaOld_16.jpg	JapaOld Akira 1763226	0.50658	1.6985	Sucesso
JapaOld_17.jpg	JapaOld Akira 1763226	0.528713	1.69545	Sucesso
JapaOld_19.jpg	JapaOld Akira 1763226	0.601655	1.71979	Sucesso
JapaOld_21.jpg	JapaOld Akira 1763226	0.58411	1.60912	Sucesso
JapaOld_22.jpg	JapaOld Akira 1763226	0.58411	1.62731	Sucesso
JapaOld_24.jpg	JapaOld Akira 1763226	0.581423	1.66135	Sucesso
JapaOld_25.jpg	JapaOld Akira 1763226	0.594037	1.71894	Sucesso
JapaOld_27.jpg	JapaOld Akira 1763226	0.581423	1.68503	Sucesso
JapaOld_29.jpg	JapaOld Akira 1763226	0.522792	1.68335	Sucesso
JapaOld_30.jpg	JapaOld Akira 1763226	0.524804	1.67672	Sucesso
JapaOld_32.jpg	JapaOld Akira 1763226	0.594037	1.68388	Sucesso
JapaOld_34.jpg	JapaOld Akira 1763226	0.538284	1.71836	Sucesso
JapaShort_0.jpg	JapaShort Okuri 1232266	0.444716	2.26572	Sucesso
JapaShort_2.jpg	JapaShort Okuri 1232266	0.446024	2.37447	Sucesso
JapaShort_6.jpg	JapaShort Okuri 1232266	0.403095	1.61846	Sucesso
JapaShort_8.jpg	JapaShort Okuri 1232266	0.399425	1.65293	Sucesso

JapaShort_9.jpg	JapaShort Okuri 1232266	0.361611	1.66548	Sucesso
JapaShort_11.jpg	JapaShort Okuri 1232266	0.384463	1.62684	Sucesso
JapaShort_13.jpg	JapaFun Nagawa 3963226	0.622365	1.61759	Falhou
JapaShort_14.jpg	JapaFun Nagawa 3963226	0.640098	1.56708	Falhou
JapaShort_16.jpg	JapaShort Okuri 1232266	0.364138	2.31895	Sucesso
JapaShort_18.jpg	JapaShort Okuri 1232266	0.461715	1.64773	Sucesso
JapaShort_20.jpg	JapaShort Okuri 1232266	0.642516	1.74762	Sucesso
JapaShort_21.jpg	JapaUS Nashira 4963226	0.652922	1.77156	Falhou
JapaShort_23.jpg	JapaShort Okuri 1232266	0.377721	1.62693	Sucesso
JapaShort_25.jpg	JapaShort Okuri 1232266	0.516346	1.61962	Sucesso
JapaShort_26.jpg	JapaShort Okuri 1232266	0.492827	2.37173	Sucesso
JapaShort_28.jpg	JapaShort Okuri 1232266	0.392135	1.67854	Sucesso
JapaShort_30.jpg	JapaShort Okuri 1232266	0.399556	1.66981	Sucesso
JapaHair_0.jpg	JapaWhite Hourish 2963226	0.53529	1.57443	Falhou
JapaHair_1.jpg	JapaWhite Hourish 2963226	0.507899	1.56557	Falhou
JapaHair_3.jpg	JapaUS Nashira 4963226	0.616936	1.50499	Falhou
JapaHair_4.jpg	JapaUS Nashira 4963226	0.525048	1.54993	Falhou
JapaHair_6.jpg	JapaWhite Hourish 2963226	0.480293	2.64575	Falhou
JapaHair_8.jpg	JapaWhite Hourish 2963226	0.598818	1.53525	Falhou
JapaHair_11.jpg	JapaShort Okuri 1232266	0.433572	1.53069	Falhou
JapaHair_12.jpg	JapaUS Nashira 4963226	0.562376	1.54957	Falhou
JapaHair_14.jpg	JapaUS Nashira 4963226	0.602046	1.53377	Falhou
JapaHair_15.jpg	JapaWhite Hourish 2963226	0.517241	2.31701	Falhou
JapaHair_19.jpg	JapaUS Nashira 4963226	0.642303	1.58624	Falhou
JapaHair_22.jpg	JapaShort Okuri 1232266	0.445537	1.6357	Falhou
JapaHair_25.jpg	JapaUS Nashira 4963226	0.527189	1.55378	Falhou
JapaHair_28.jpg	JapaUS Nashira 4963226	0.6245	2.26375	Falhou
JapaLong_2.jpg	Unknown	0.700863	1.24294	Falhou
JapaLong_3.jpg	Unknown	0.702915	1.9467	Falhou
JapaLong_6.jpg	Unknown	0.744095	1.92303	Falhou
JapaLong_8.jpg	Unknown	0.763297	1.27018	Falhou
JapaLong_9.jpg	Unknown	0.716258	1.24699	Falhou
JapaLong_10.jpg	JapaLong Oishi 1232257	0.664556	1.56306	Sucesso
JapaLong_13.jpg	JapaLong Oishi 1232257	0.656752	1.55237	Sucesso
JapaLong_14.jpg	JapaLong Oishi 1232257	0.639545	2.21207	Sucesso
JapaLong_18.jpg	JapaLong Oishi 1232257	0.563845	1.6076	Sucesso
JapaLong_21.jpg	JapaLong Oishi 1232257	0.692666	1.59806	Sucesso
JapaLong_23.jpg	Unknown	0.729627	1.22096	Falhou
JapaLong_24.jpg	Unknown	0.775852	1.23587	Falhou
JapaLong_25.jpg	Unknown	0.775852	1.25425	Falhou
JapaLong_26.jpg	JapaLong Oishi 1232257	0.646453	1.57442	Sucesso
JapaLong_28.jpg	Unknown	0.744095	1.27409	Falhou
JapaLong_29.jpg	JapaLong Oishi 1232257	0.549622	1.61371	Sucesso
JapaLong_31.jpg	JapaLong Oishi 1232257	0.692666	1.57511	Sucesso

Média	0.56	1.69
Desvio Padrão	0.11	0.30
Total Amostras	290	

Falhou	23.8%
Sucesso	75.9%

APÊNDICE E - Tabelas testes do algoritmo de detecção com *LBP*

Tablea 1: Threshold = 0.5

Arquivo de Entrada	Nome Identificado	Similaridade	Tempo(s)	Resultado
Robson_1.jpg	Robson Silva 1111114	0.374507	1.30292	Sucesso
Robson_2.jpg	Robson Silva 1111114	0.380971	1.30495	Sucesso
Robson_4.jpg	Robson Silva 1111114	0.38645	1.26805	Sucesso
Robson_5.jpg	Robson Silva 1111114	0.398453	1.29862	Sucesso
Robson_6.jpg	Robson Silva 1111114	0.364721	1.53448	Sucesso
Robson_8.jpg	Robson Silva 1111114	0.440192	1.30972	Sucesso
Robson_9.jpg	Robson Silva 1111114	0.429284	1.23574	Sucesso
Robson_10.jpg	Robson Silva 1111114	0.467387	1.25811	Sucesso
Robson_11.jpg	Robson Silva 1111114	0.325347	1.31893	Sucesso
Robson_13.jpg	Robson Silva 1111114	0.300655	1.29883	Sucesso
Robson_14.jpg	Robson Silva 1111114	0.342467	1.29718	Sucesso
Robson_15.jpg	Robson Silva 1111114	0.382108	1.2889	Sucesso
Robson_16.jpg	Robson Silva 1111114	0.302352	1.29753	Sucesso
Robson_18.jpg	Robson Silva 1111114	0.464708	1.20942	Sucesso
Robson_19.jpg	Unknown	0.536395	0.940187	Sucesso
Robson_20.jpg	Robson Silva 1111114	0.315979	1.27323	Falhou
Robson_21.jpg	Robson Silva 1111114	0.414376	1.29272	Sucesso
Robson_22.jpg	Robson Silva 1111114	0.42418	1.31748	Sucesso
Robson_24.jpg	Robson Silva 1111114	0.361241	1.29033	Sucesso
Robson_25.jpg	Robson Silva 1111114	0.471257	1.30466	Sucesso
Robson_26.jpg	Robson Silva 1111114	0.402217	1.26968	Sucesso
Robson_27.jpg	Robson Silva 1111114	0.343784	1.30572	Sucesso
Robson_29.jpg	Robson Silva 1111114	0.489488	1.22665	Sucesso
Robson_31.jpg	Robson Silva 1111114	0.439782	1.30119	Sucesso
Robson_32.jpg	Robson Silva 1111114	0.451032	1.3323	Sucesso
Robson_34.jpg	Robson Silva 1111114	0.42418	1.29346	Sucesso
Nune_0.jpg	Unknown	0.742087	0.831495	Sucesso
Nune_1.jpg	Unknown	0.728553	1.56737	Falhou
Nune_2.jpg	Unknown	0.748399	0.818748	Falhou
Nune_3.jpg	Unknown	0.791909	0.874523	Falhou
Nune_4.jpg	Unknown	0.741905	0.81703	Falhou
Nune_5.jpg	Unknown	0.630136	0.858035	Falhou
Nune_6.jpg	Unknown	0.746968	0.874564	Falhou
Nune_7.jpg	Unknown	0.664439	0.885953	Falhou
Nune_8.jpg	Unknown	0.708895	1.36781	Falhou
Nune_9.jpg	Unknown	0.654761	0.99118	Falhou
Nune_10.jpg	Unknown	0.54204	1.67032	Falhou
Nune_12.jpg	Unknown	0.61162	0.874474	Falhou
Nune_13.jpg	Unknown	0.553149	0.838584	Falhou
Nune_14.jpg	Unknown	0.524975	0.958779	Falhou
Nune_15.jpg	Unknown	0.505763	0.897385	Falhou
Nune_16.jpg	Unknown	0.526663	0.876924	Falhou
Nune_17.jpg	Unknown	0.531909	0.966516	Falhou
Nune_18.jpg	Unknown	0.728553	0.887128	Falhou
Nune_19.jpg	Unknown	0.791909	0.892819	Falhou
Nune_20.jpg	Unknown	0.726935	0.881835	Falhou
Nune_21.jpg	Nune nascimento 1111112	0.472558	1.17022	Falhou
Nune_22.jpg	Unknown	0.537245	0.916067	Sucesso
Nune_23.jpg	Unknown	0.689223	0.860772	Falhou
Nune_24.jpg	Unknown	0.804774	0.848482	Falhou
Nune_26.jpg	Unknown	0.624868	1.04664	Falhou
Nune_27.jpg	Unknown	0.752905	0.88295	Falhou
Nune_30.jpg	Unknown	0.61612	1.13345	Falhou
Nune_31.jpg	Unknown	0.728553	0.966224	Falhou
Nune_32.jpg	Unknown	0.664439	0.852507	Falhou
Nune_33.jpg	Unknown	0.630136	0.875868	Falhou
Nune_34.jpg	Unknown	0.791909	0.897868	Falhou
Lucas_0.jpg	Lucas Telles 1111111	0.419685	1.4516	Sucesso
Lucas_1.jpg	Unknown	0.633972	0.71016	Sucesso
Lucas_2.jpg	Unknown	0.574719	1.01887	Falhou
Lucas_3.jpg	Unknown	0.535454	0.998484	Falhou
Lucas_4.jpg	Unknown	0.571867	1.01776	Falhou
Lucas_5.jpg	Unknown	0.575558	0.995417	Falhou
Lucas_6.jpg	Unknown	0.604901	0.992416	Falhou
Lucas_7.jpg	Unknown	0.567795	0.97063	Falhou
Lucas_8.jpg	Unknown	0.583176	0.976347	Falhou
Lucas_9.jpg	Unknown	0.633202	0.16169	Falhou
Lucas_10.jpg	Unknown	0.531642	1.5425	Falhou
Lucas_12.jpg	Unknown	0.530283	1.23038	Falhou
Lucas_13.jpg	Unknown	0.511365	1.00693	Falhou
Lucas_14.jpg	Unknown	0.604388	0.959757	Falhou
Lucas_15.jpg	Unknown	0.524454	0.961921	Falhou
Lucas_16.jpg	Unknown	0.507858	1.40028	Falhou
Lucas_17.jpg	Lucas Telles 1111111	0.372664	1.48252	Falhou
Lucas_18.jpg	Unknown	0.62808	0.993426	Sucesso
Lucas_19.jpg	Unknown	0.572806	0.973301	Falhou
Lucas_20.jpg	Lucas Telles 1111111	0.389479	1.3477	Falhou
Lucas_21.jpg	Unknown	0.500076	0.98541	Sucesso
Lucas_22.jpg	Lucas Telles 1111111	0.498972	1.2859	Falhou
Lucas_24.jpg	Lucas Telles 1111111	0.445803	1.29416	Sucesso
Lucas_25.jpg	Lucas Telles 1111111	0.445803	1.28743	Sucesso
Lucas_26.jpg	Lucas Telles 1111111	0.417575	1.34897	Sucesso
Lucas_27.jpg	Unknown	0.583096	0.9553	Sucesso
Lucas_28.jpg	Lucas Telles 1111111	0.439836	1.33677	Falhou
Lucas_30.jpg	Unknown	0.65803	0.974702	Sucesso
Lucas_31.jpg	Lucas Telles 1111111	0.454663	1.31676	Falhou
Lucas_32.jpg	Unknown	0.66109	0.984727	Sucesso
Lucas_33.jpg	Unknown	0.500158	0.922823	Falhou
Lucas_34.jpg	Lucas Telles 1111111	0.42429	1.30815	Falhou
Rafael_0.jpg	Unknown	0.690679	0.957674	Sucesso
Rafael_1.jpg	Unknown	0.543035	1.57901	Falhou
Rafael_2.jpg	Unknown	0.52483	1.03715	Falhou
Rafael_3.jpg	Unknown	0.62056	0.968049	Falhou
Rafael_4.jpg	Rafael Ribeiro 1111113	0.499806	1.31263	Falhou

Rafael_6.jpg	Rafael Ribeiro 1111113	0.494571	2.02118	Sucesso
Rafael_8.jpg	Unknown	0.535734	1.01763	Sucesso
Rafael_9.jpg	Unknown	0.566457	0.952976	Falhou
Rafael_10.jpg	Unknown	0.546249	0.937894	Falhou
Rafael_11.jpg	Rafael Ribeiro 1111113	0.499559	1.91012	Falhou
Rafael_13.jpg	Unknown	0.532825	0.983612	Sucesso
Rafael_14.jpg	Unknown	0.538901	1.1606	Falhou
Rafael_15.jpg	Unknown	0.504379	0.959701	Falhou
Rafael_16.jpg	Unknown	0.502877	0.996226	Falhou
Rafael_17.jpg	Unknown	0.555504	0.981765	Falhou
Rafael_18.jpg	Unknown	0.531744	0.967968	Falhou
Rafael_19.jpg	Unknown	0.606704	0.944344	Falhou
Rafael_20.jpg	Unknown	0.648147	0.949623	Falhou
Rafael_21.jpg	Unknown	0.554632	0.965742	Falhou
Rafael_22.jpg	Unknown	0.614761	1.38476	Falhou
Rafael_23.jpg	Unknown	0.564083	1.17006	Falhou
Rafael_24.jpg	Unknown	0.561799	0.978547	Falhou
Rafael_25.jpg	Unknown	0.67084	0.996802	Falhou
Rafael_26.jpg	Unknown	0.632511	0.988952	Falhou
Rafael_27.jpg	Unknown	0.572162	0.995997	Falhou
Rafael_28.jpg	Unknown	0.625572	0.964816	Falhou
Rafael_29.jpg	Unknown	0.642632	0.961393	Falhou
Rafael_30.jpg	Unknown	0.573085	0.964195	Falhou
Rafael_31.jpg	Unknown	0.648147	0.948685	Falhou
Rafael_32.jpg	Unknown	0.566457	0.979359	Falhou
Rafael_33.jpg	Unknown	0.611468	1.67317	Falhou
Rafael_34.jpg	Unknown	0.62056	0.977048	Falhou
Japa_0.jpg	Japa Kim 1622211	0.346483	1.2705	Falhou
Japa_1.jpg	Japa Kim 1622211	0.382731	1.88787	Sucesso
Japa_3.jpg	Japa Kim 1622211	0.288107	1.23942	Sucesso
Japa_4.jpg	Japa Kim 1622211	0.381367	1.31822	Sucesso
Japa_6.jpg	Japa Kim 1622211	0.352834	1.2641	Sucesso
Japa_7.jpg	Japa Kim 1622211	0.367248	1.98706	Sucesso
Japa_9.jpg	Japa Kim 1622211	0.361658	1.41206	Sucesso
Japa_10.jpg	Japa Kim 1622211	0.379294	1.2996	Sucesso
Japa_11.jpg	Japa Kim 1622211	0.438586	1.93385	Sucesso
Japa_14.jpg	Japa Kim 1622211	0.37006	1.24975	Sucesso
Japa_15.jpg	Japa Kim 1622211	0.392447	1.26077	Sucesso
Japa_16.jpg	Japa Kim 1622211	0.411494	1.30604	Sucesso
Japa_17.jpg	Japa Kim 1622211	0.421273	2.0182	Sucesso
Japa_19.jpg	Japa Kim 1622211	0.427177	1.25075	Sucesso
Japa_20.jpg	Japa Kim 1622211	0.449316	1.26579	Sucesso
Japa_22.jpg	Japa Kim 1622211	0.416085	1.2719	Sucesso
Japa_23.jpg	Japa Kim 1622211	0.414364	1.27609	Sucesso
Japa_24.jpg	Japa Kim 1622211	0.375422	1.27281	Sucesso
Japa_25.jpg	Japa Kim 1622211	0.392222	1.30047	Sucesso
Japa_26.jpg	Japa Kim 1622211	0.370105	1.30117	Sucesso
Japa_28.jpg	Japa Kim 1622211	0.370145	1.90147	Sucesso
Japa_29.jpg	Japa Kim 1622211	0.355575	1.30835	Sucesso
Japa_30.jpg	Japa Kim 1622211	0.313288	1.27642	Sucesso
Japa_32.jpg	Japa Kim 1622211	0.383826	1.24949	Sucesso
JapaMan_0.jpg	Unknown	0.502933	1.02087	Sucesso
JapaMan_1.jpg	JapaMan KimK 1234567	0.47248	1.32215	Falhou
JapaMan_2.jpg	Unknown	0.596395	0.975127	Sucesso
JapaMan_3.jpg	JapaMan KimK 1234567	0.410742	1.35315	Falhou
JapaMan_4.jpg	Unknown	0.530969	0.96856	Sucesso
JapaMan_5.jpg	Unknown	0.529443	1.03929	Falhou
JapaMan_8.jpg	Unknown	0.56011	1.04299	Falhou
JapaMan_9.jpg	JapaMan KimK 1234567	0.461039	1.33878	Falhou
JapaMan_10.jpg	JapaMan KimK 1234567	0.432079	2.00466	Sucesso
JapaMan_12.jpg	JapaMan KimK 1234567	0.390608	1.33943	Sucesso
JapaMan_13.jpg	JapaMan KimK 1234567	0.418638	1.92721	Sucesso
JapaMan_15.jpg	Unknown	0.556205	1.01168	Sucesso
JapaMan_18.jpg	JapaMan KimK 1234567	0.463739	1.38041	Falhou
JapaMan_19.jpg	JapaMan KimK 1234567	0.487505	1.31638	Sucesso
JapaMan_20.jpg	JapaMan KimK 1234567	0.449195	1.97133	Sucesso
JapaMan_22.jpg	Unknown	0.519585	1.62963	Sucesso
JapaMan_23.jpg	Unknown	0.552039	0.970048	Falhou
JapaMan_24.jpg	Unknown	0.511565	1.00519	Falhou
JapaMan_25.jpg	JapaMan KimK 1234567	0.497703	1.34302	Falhou
JapaMan_29.jpg	JapaMan KimK 1234567	0.378683	1.50863	Sucesso
JapaMan_30.jpg	JapaMan KimK 1234567	0.450354	1.85261	Sucesso
JapaMan_32.jpg	JapaMan KimK 1234567	0.410898	1.4086	Sucesso
JapaChild_0.jpg	Unknown	0.582624	0.91153	Sucesso
JapaChild_1.jpg	Unknown	0.568559	0.926081	Falhou
JapaChild_2.jpg	Unknown	0.519605	0.871456	Falhou
JapaChild_3.jpg	Unknown	0.582624	1.49492	Falhou
JapaChild_4.jpg	JapaChild Assaki 14132266	0.244439	1.15896	Falhou
JapaChild_5.jpg	Unknown	0.601061	0.898142	Sucesso
JapaChild_6.jpg	Unknown	0.582624	0.897824	Falhou
JapaChild_7.jpg	Unknown	0.642181	0.935585	Falhou
JapaChild_8.jpg	Unknown	0.59337	1.06901	Falhou
JapaChild_9.jpg	Unknown	0.601061	0.956954	Falhou
JapaChild_10.jpg	Unknown	0.551592	0.895456	Falhou
JapaChild_11.jpg	Unknown	0.551592	0.896928	Falhou
JapaChild_12.jpg	JapaChild Assaki 14132266	0.36469	1.11796	Falhou
JapaChild_13.jpg	Unknown	0.583093	2.26081	Sucesso
JapaChild_15.jpg	Unknown	0.627366	1.05098	Falhou
JapaChild_16.jpg	Unknown	0.526704	0.881702	Falhou
JapaChild_17.jpg	Unknown	0.564543	0.898791	Falhou</td

JapaChild_26.jpg	Unknown	0.583093	0.932293	Falhou
JapaChild_28.jpg	Unknown	0.642181	0.934786	Falhou
JapaChild_29.jpg	Unknown	0.524968	1.56706	Falhou
JapaChild_31.jpg	Unknown	0.514311	0.876247	Falhou
JapaChild_32.jpg	Unknown	0.537098	0.89023	Falhou
JapaChild_33.jpg	JapaChild Assaki 14132266	0.472778	1.13206	Falhou
JapaChild_34.jpg	Unknown	0.582624	1.03832	Sucesso
JapaNew_0.jpg	Unknown	0.667111	0.901042	Falhou
JapaNew_1.jpg	Unknown	0.643172	1.07588	Falhou
JapaNew_2.jpg	Unknown	0.650356	0.90089	Falhou
JapaNew_3.jpg	Unknown	0.630966	0.864878	Falhou
JapaNew_4.jpg	Unknown	0.630966	0.951944	Falhou
JapaNew_5.jpg	Unknown	0.51853	1.48105	Falhou
JapaNew_7.jpg	Unknown	0.581337	0.923318	Falhou
JapaNew_8.jpg	Unknown	0.581337	1.56234	Falhou
JapaNew_10.jpg	Unknown	0.636606	1.03394	Falhou
JapaNew_11.jpg	Unknown	0.667111	0.883709	Falhou
JapaNew_12.jpg	Unknown	0.673094	0.860762	Falhou
JapaNew_13.jpg	Unknown	0.639006	1.50074	Falhou
JapaNew_15.jpg	Unknown	0.522185	1.06297	Falhou
JapaNew_16.jpg	Unknown	0.545831	0.93271	Falhou
JapaNew_17.jpg	Unknown	0.563366	0.892476	Falhou
JapaNew_18.jpg	Unknown	0.563366	0.875124	Falhou
JapaNew_19.jpg	Unknown	0.598778	1.00318	Falhou
JapaNew_20.jpg	Unknown	0.598778	0.908092	Falhou
JapaNew_21.jpg	Unknown	0.637358	0.933162	Falhou
JapaNew_22.jpg	Unknown	0.624964	0.908224	Falhou
JapaNew_23.jpg	Unknown	0.642381	0.846587	Falhou
JapaNew_24.jpg	Unknown	0.630207	1.11914	Falhou
JapaNew_25.jpg	JapaNew Assaki 1223226	0.459953	1.19642	Falhou
JapaNew_26.jpg	Unknown	0.626095	0.901382	Sucesso
JapaNew_27.jpg	Unknown	0.641934	0.893984	Falhou
JapaNew_28.jpg	Unknown	0.629369	0.891276	Falhou
JapaNew_29.jpg	Unknown	0.629369	1.53183	Falhou
JapaNew_30.jpg	Unknown	0.522098	0.930167	Falhou
JapaNew_31.jpg	Unknown	0.545831	0.952357	Falhou
JapaNew_32.jpg	Unknown	0.650356	0.902091	Falhou
JapaNew_33.jpg	Unknown	0.643172	1.07686	Falhou
JapaNew_34.jpg	Unknown	0.545831	1.4375	Falhou
JapaFat_0.jpg	Unknown	0.602244	1.06317	Falhou
JapaFat_2.jpg	Unknown	0.602244	0.904218	Sucesso
JapaFat_3.jpg	Unknown	0.561051	0.941606	Sucesso
JapaFat_6.jpg	Unknown	0.564662	0.957964	Sucesso
JapaFat_7.jpg	Unknown	0.564662	0.95958	Sucesso
JapaFat_8.jpg	Unknown	0.564662	1.955987	Sucesso
JapaFat_9.jpg	Unknown	0.566324	1.45586	Sucesso
JapaFat_10.jpg	Unknown	0.576888	1.08438	Sucesso
JapaFat_11.jpg	Unknown	0.576888	0.919583	Sucesso
JapaFat_12.jpg	JapaNew Assaki 1223226	0.493614	1.21856	Sucesso
JapaFat_13.jpg	Unknown	0.513193	0.930109	Falhou
JapaFat_14.jpg	JapaNew Assaki 1223226	0.481384	1.90124	Sucesso
JapaFat_16.jpg	JapaNew Assaki 1223226	0.481384	1.24231	Falhou
JapaFat_17.jpg	Unknown	0.645136	0.940655	Falhou
JapaFat_18.jpg	Unknown	0.547243	0.94124	Sucesso
JapaFat_19.jpg	Unknown	0.594799	0.933383	Sucesso
JapaFat_20.jpg	Unknown	0.594799	0.92116	Sucesso
JapaFat_21.jpg	Unknown	0.56793	0.912961	Sucesso
JapaFat_22.jpg	Unknown	0.56793	1.01671	Sucesso
JapaFat_23.jpg	Unknown	0.570267	0.928999	Sucesso
JapaFat_24.jpg	JapaNew Assaki 1223226	0.486974	1.80745	Sucesso
JapaFat_25.jpg	JapaNew Assaki 1223226	0.458574	1.43259	Falhou
JapaFat_27.jpg	JapaNew Assaki 1223226	0.458574	1.98101	Falhou
JapaFat_29.jpg	Unknown	0.615522	0.921984	Falhou
JapaFat_30.jpg	Unknown	0.615522	1.73805	Sucesso
JapaFat_31.jpg	Unknown	0.554089	0.922884	Sucesso
JapaFat_32.jpg	Unknown	0.62542	0.937217	Sucesso
JapaFat_33.jpg	Unknown	0.62542	0.934434	Sucesso
JapaFat_34.jpg	Unknown	0.62542	0.942373	Sucesso
JapaUS_0.jpg	Unknown	0.678534	1.19219	Sucesso
JapaUS_1.jpg	Unknown	0.707943	1.10754	Falhou
JapaUS_2.jpg	Unknown	0.656741	0.971965	Falhou
JapaUS_3.jpg	Unknown	0.656741	0.968817	Falhou
JapaUS_4.jpg	Unknown	0.655285	0.982323	Falhou
JapaUS_5.jpg	Unknown	0.655285	1.51249	Falhou
JapaUS_6.jpg	Unknown	0.617153	1.10237	Falhou
JapaUS_7.jpg	Unknown	0.625901	0.982175	Falhou
JapaUS_8.jpg	Unknown	0.644889	0.98305	Falhou
JapaUS_9.jpg	Unknown	0.598045	1.00706	Falhou
JapaUS_10.jpg	Unknown	0.692207	1.10816	Falhou
JapaUS_11.jpg	Unknown	0.712927	1.00816	Falhou
JapaUS_12.jpg	Unknown	0.635101	0.986839	Falhou
JapaUS_13.jpg	Unknown	0.659283	0.930556	Falhou
JapaUS_14.jpg	Unknown	0.656294	0.977785	Falhou
JapaUS_15.jpg	Unknown	0.666695	1.5144	Falhou
JapaUS_17.jpg	Unknown	0.69603	1.15482	Falhou
JapaUS_18.jpg	Unknown	0.666497	0.978722	Falhou
JapaUS_19.jpg	Unknown	0.650085	1.00043	Falhou
JapaUS_20.jpg	Unknown	0.697112	2.18883	Falhou
JapaUS_22.jpg	Unknown	0.71176	0.973052	Falhou
JapaUS_23.jpg	Unknown	0.701844	0.963648	Falhou
JapaUS_24.jpg	Unknown	0.60803	0.962985	Falhou
JapaUS_25.jpg	Unknown	0.642892	0.939334	Falhou
JapaUS_26.jpg	Unknown	0.605552	0.987096	Falhou
JapaUS_27.jpg	Unknown	0.707943	1.53983	Falhou
JapaUS_28.jpg	Unknown	0.70670	1.14962	Falhou
JapaUS_29.jpg	Unknown	0.678534	1.00757	Falhou
JapaUS_30.jpg	Unknown	0.612936	1.01733	Falhou
JapaUS_31.jpg	Unknown	0.666695	1.00256	Falhou
JapaUS_32.jpg	Unknown	0.692207	1.45608	Falhou
JapaUS_33.jpg	Unknown	0.617153	0.98044	Falhou
JapaUS_34.jpg	Unknown	0.71176	0.966167	Falhou
JapaCN_0.jpg	JapaCN Nawan 7963226	0.462851	1.27816	Falhou
JapaCN_1.jpg	Unknown	0.58745	0.964485	Sucesso
JapaCN_2.jpg	JapaCN Nawan 7963226	0.491287	1.27588	Falhou
JapaCN_3.jpg	Unknown	0.591459	0.966543	Sucesso

JapaCN_4.jpg	Unknown	0.505087	0.958434	Falhou
JapaCN_5.jpg	Unknown	0.526742	0.977623	Falhou
JapaCN_6.jpg	JapaCN Nawan 7963226	0.468455	1.31702	Falhou
JapaCN_8.jpg	Unknown	0.570348	1.58548	Sucesso
JapaCN_9.jpg	Unknown	0.642077	0.982329	Falhou
JapaCN_10.jpg	Unknown	0.61298	0.953726	Falhou
JapaCN_11.jpg	Unknown	0.663061	0.975755	Falhou
JapaCN_12.jpg	Unknown	0.67156	0.996269	Falhou
JapaCN_13.jpg	JapaCN Nawan 7963226	0.467957	1.34363	Falhou
JapaCN_14.jpg	Unknown	0.491287	1.27296	Sucesso
JapaCN_15.jpg	Unknown	0.626222	0.962655	Sucesso
JapaCN_16.jpg	Unknown	0.633809	0.980165	Falhou
JapaCN_17.jpg	Unknown	0.659816	0.972238	Falhou
JapaCN_18.jpg	JapaCN Nawan 7963226	0.475576	1.44077	Falhou
JapaCN_19.jpg	Unknown	0.680918	0.939254	Sucesso
JapaCN_20.jpg	Unknown	0.674688	0.96113	Falhou
JapaCN_21.jpg	Unknown	0.683943	0.958199	Falhou
JapaCN_22.jpg	Unknown	0.725732	0.878834	Falhou
JapaCN_23.jpg	Unknown	0.626222	1.33837	Falhou
JapaCN_24.jpg	Unknown	0.621082	1.10594	Falhou
JapaCN_25.jpg	Unknown	0.650822	0.955017	Falhou
JapaCN_26.jpg	Unknown	0.594029	0.961226	Falhou
JapaCN_27.jpg	JapaCN Nawan 7963226	0.462851	1.30255	Falhou
JapaCN_29.jpg	Unknown	0.703544	0.56788	Sucesso
JapaCN_30.jpg	Unknown	0.562231	1.11902	Falhou
JapaCN_31.jpg	Unknown	0.683943	1.79709	Falhou
JapaCN_33.jpg	Unknown	0.621082	0.945679	Falhou
JapaCN_34.jpg	Unknown	0.659816	0.998744	Falhou
JapaWhite_0.jpg	JapaWhite Houriish 2963226	0.432055	1.29305	Falhou
JapaWhite_2.jpg	JapaWhite Houriish 2963226	0.460985	1.24762	Sucesso
JapaWhite_3.jpg	JapaWhite Houriish 2963226	0.400579	1.24881	Sucesso
JapaWhite_4.jpg	JapaWhite Houriish 2963226	0.44452	1.36058	Sucesso
JapaWhite_5.jpg	JapaWhite Houriish 2963226	0.48446	1.2765	Sucesso
JapaWhite_7.jpg	JapaWhite Houriish 2963226	0.465976	1.26051	Sucesso
JapaWhite_8.jpg	JapaWhite Houriish 2963226	0.367916	1.26027	Sucesso
JapaWhite_9.jpg	JapaWhite Houriish 2963226	0.371752	1.25487	Sucesso
JapaWhite_10.jpg	JapaWhite Houriish 2963226	0.371752	1.25564	Sucesso
JapaWhite_12.jpg	JapaWhite Houriish 2963226	0.435674	1.25619	Sucesso
JapaWhite_14.jpg	JapaWhite Houriish 2963226	0.432055	2.60982	Sucesso
JapaWhite_16.jpg	JapaWhite Houriish 2963226	0.415402	1.27989	Sucesso
JapaWhite_18.jpg	JapaWhite Houriish 2963226	0.313192	1.48316	Sucesso
JapaWhite_20.jpg	JapaWhite Houriish 2963226	0.485527	1.23202	Sucesso
JapaWhite_21.jpg	JapaWhite Houriish 2963226	0.496042	1.25631	Sucesso
JapaWhite_22.jpg	JapaWhite Houriish 2963226	0.432055	1.25704	Sucesso
JapaWhite_23.jpg	JapaWhite Houriish 2963226	0.399878	1.89413	Sucesso
JapaWhite_25.jpg	JapaWhite Houriish 2963226	0.472319	1.38083	Sucesso
JapaWhite_26.jpg	JapaWhite Houriish 2963226	0.399878	1.27253	Sucesso
JapaWhite_27.jpg	JapaWhite Houriish 2963226	0.313192	1.48843	Sucesso
JapaWhite_29.jpg	JapaWhite Houriish 2963226	0.484005	1.26297	Sucesso
JapaWhite_30.jpg	JapaWhite Houriish 2963226	0.400579	1.24923	Sucesso
JapaWhite_31.jpg	JapaWhite Houriish 2963226	0.472319	1.25732	Sucesso
JapaWhite_32.jpg	JapaWhite Houriish 2963226	0.248195	1.27645	Sucesso
JapaWhite_34.jpg	JapaWhite Houriish 2963226	0.248195	1.26645	Sucesso
JapaOld_0.jpg	Unknown	0.529076	1.10925	Sucesso
JapaOld_1.jpg	JapaOld Akira 1763226	0.490306	1.33611	Falhou
JapaOld_2.jpg	Unknown	0.532657	1.00776	Sucesso
JapaOld_3.jpg	JapaOld Akira 1763226	0.471977	1.34574	Falhou
JapaOld_5.jpg	JapaOld Akira 1763226	0.397407	1.24842	Sucesso
JapaOld_6.jpg	Unknown	0.532657	0.987903	Sucesso
JapaOld_7.jpg	JapaOld Akira 1763226	0.493578	1.34196	Falhou
JapaOld_8.jpg	JapaOld Akira 1763226	0.486256	1.37403	Sucesso
JapaOld_9.jpg	Unknown	0.51115	1.03213	Sucesso
JapaOld_10.jpg	Unknown	0.511844	1.04659	Falhou
JapaOld_11.jpg	Unknown	0.524908	1.00376	Falhou
JapaOld_12.jpg	JapaOld Akira 1763226	0.494006	1.36107	Falhou
JapaOld_14.jpg	Unknown	0.519614	1.35692	Sucesso
JapaOld_15.jpg	Unknown	0.529076	1.28602	Falhou
JapaOld_16.jpg	JapaOld Akira 1763226	0.471977	1.35015	Falhou
JapaOld_18.jpg	JapaOld Akira 1763226	0.485922	1.36589	Sucesso
JapaOld_19.jpg	Unknown	0.511546	1.00196	Sucesso
JapaOld_20.jpg	Unknown	0.530487	1.00708	Falhou
JapaOld_21.jpg	Unknown	0.561856	1.00289	Falhou
JapaOld_22.jpg	Unknown	0.561856	0.999281	Falhou
JapaOld_23.jpg	Unknown	0.521134	1.04153	Falhou
JapaOld_24.jpg	Unknown	0.566533	1.73739	Falhou
JapaOld_27.jpg	Unknown	0.566533	1.06127	Falhou
JapaOld_28.jpg	Unknown	0.535629	1.04639	Falhou
JapaOld_29.jpg	Unknown	0.545888	1.54096	Falhou
JapaOld_30.jpg	Unknown	0.529076	1.26463	Falhou
JapaOld_32.jpg	Unknown	0.590473	1.03366	Falhou
JapaOld_33.jpg	Unknown	0.521134	1.042	Falhou
JapaOld_34.jpg	Unknown	0.519614	1.02383	Falhou
JapaShort_0.jpg	JapaShort Okuri 1232266	0.443096	1.7569	Falhou
JapaShort_1.jpg	JapaShort Okuri 1232266	0.422615	1.29951	Sucesso
JapaShort_2.jpg	JapaShort Okuri 1232266	0.4219		

JapaShort_31.jpg	JapaShort Okuri 1232266	0.443096	1.94659	Sucesso
JapaHair_1.jpg	Unknown	0.626232	0.943129	Sucesso
JapaHair_3.jpg	Unknown	0.583009	0.972	Sucesso
JapaHair_4.jpg	Unknown	0.574655	0.963165	Sucesso
JapaHair_5.jpg	Unknown	0.508948	0.929256	Sucesso
JapaHair_6.jpg	JapaWhite Houriish 2963226	0.395863	1.25669	Sucesso
JapaHair_8.jpg	JapaWhite Houriish 2963226	0.423955	1.23816	Falhou
JapaHair_9.jpg	JapaWhite Houriish 2963226	0.433921	1.23283	Falhou
JapaHair_10.jpg	Unknown	0.537979	0.931257	Falhou
JapaHair_11.jpg	Unknown	0.654009	1.06453	Sucesso
JapaHair_12.jpg	Unknown	0.632239	1.00349	Sucesso
JapaHair_13.jpg	Unknown	0.623511	0.971356	Sucesso
JapaHair_14.jpg	Unknown	0.602046	0.936466	Sucesso
JapaHair_15.jpg	Unknown	0.609358	0.926761	Sucesso
JapaHair_16.jpg	Unknown	0.550213	0.931114	Sucesso
JapaHair_17.jpg	Unknown	0.641339	1.3104	Sucesso
JapaHair_18.jpg	JapaWhite Houriish 2963226	0.495517	1.35959	Sucesso
JapaHair_19.jpg	Unknown	0.641426	0.942261	Falhou
JapaHair_20.jpg	Unknown	0.609417	1.56462	Sucesso
JapaHair_23.jpg	JapaShort Okuri 1232266	0.355387	1.34843	Sucesso
JapaHair_24.jpg	JapaWhite Houriish 2963226	0.462563	1.22194	Falhou
JapaHair_25.jpg	JapaWhite Houriish 2963226	0.488596	1.22967	Falhou
JapaHair_26.jpg	Unknown	0.521169	0.967671	Falhou
JapaHair_28.jpg	Unknown	0.657668	0.966979	Sucesso
JapaLong_0.jpg	Unknown	0.735998	0.98897	Sucesso
JapaLong_1.jpg	Unknown	0.754843	0.978018	Falhou
JapaLong_2.jpg	Unknown	0.72343	1.00348	Falhou
JapaLong_3.jpg	Unknown	0.586901	1.04835	Falhou
JapaLong_6.jpg	Unknown	0.771221	1.02967	Falhou
JapaLong_7.jpg	Unknown	0.701866	0.97332	Falhou

JapaLong_8.jpg	Unknown	0.782755	1.45386	Falhou
JapaLong_9.jpg	Unknown	0.828582	1.14631	Falhou
JapaLong_10.jpg	Unknown	0.75865	0.976304	Falhou
JapaLong_11.jpg	Unknown	0.560491	0.985363	Falhou
JapaLong_12.jpg	Unknown	0.68782	0.907829	Falhou
JapaLong_13.jpg	Unknown	0.548809	1.01764	Falhou
JapaLong_14.jpg	Unknown	0.56448	0.95223	Falhou
JapaLong_15.jpg	Unknown	0.569404	0.948698	Falhou
JapaLong_16.jpg	Unknown	0.528762	0.947358	Falhou
JapaLong_17.jpg	Unknown	0.573949	0.934464	Falhou
JapaLong_21.jpg	Unknown	0.763377	1.07363	Falhou
JapaLong_22.jpg	Unknown	0.834504	0.952392	Falhou
JapaLong_23.jpg	Unknown	0.75984	0.961202	Falhou
JapaLong_24.jpg	Unknown	0.759223	0.95263	Falhou
JapaLong_25.jpg	Unknown	0.759223	0.89731	Falhou
JapaLong_26.jpg	Unknown	0.640719	0.955793	Falhou
JapaLong_27.jpg	Unknown	0.652067	0.941904	Falhou
JapaLong_28.jpg	Unknown	0.771221	0.992921	Falhou
JapaLong_29.jpg	Unknown	0.569404	0.95962	Falhou
JapaLong_30.jpg	Unknown	0.569404	0.960292	Falhou
JapaLong_31.jpg	Unknown	0.763377	0.997296	Falhou
JapaLong_32.jpg	Unknown	0.68782	0.878639	Falhou
Média		0.54	1.16	Falhou
Desvio Padrão		0.11	0.28	
Total Amostras		451		
				Falhou 60.4%
				Sucesso 39.6%

Tablea 2: Threshold = 0.6

Arquivo de Entrada	Nome Identificado	Similaridade	Tempo(s)	Resultado
Robson_0.jpg	Robson Silva 1111114	0.363329	1.37364	Sucesso
Robson_1.jpg	Robson Silva 1111114	0.374507	1.28792	Sucesso
Robson_2.jpg	Robson Silva 1111114	0.380971	1.28377	Sucesso
Robson_3.jpg	Robson Silva 1111114	0.379333	1.33295	Sucesso
Robson_5.jpg	Robson Silva 1111114	0.398453	1.29895	Sucesso
Robson_6.jpg	Robson Silva 1111114	0.364721	1.51766	Sucesso
Robson_7.jpg	Robson Silva 1111114	0.33219	1.31071	Sucesso
Robson_9.jpg	Robson Silva 1111114	0.429284	1.25558	Sucesso
Robson_10.jpg	Robson Silva 1111114	0.467387	1.28817	Sucesso
Robson_11.jpg	Robson Silva 1111114	0.325347	1.42031	Sucesso
Robson_13.jpg	Robson Silva 1111114	0.300655	1.26944	Sucesso
Robson_14.jpg	Robson Silva 1111114	0.342467	1.27019	Sucesso
Robson_15.jpg	Robson Silva 1111114	0.382108	1.27986	Sucesso
Robson_16.jpg	Robson Silva 1111114	0.302352	1.30858	Sucesso
Robson_18.jpg	Robson Silva 1111114	0.464708	1.21086	Sucesso
Robson_19.jpg	Robson Silva 1111114	0.530395	1.27244	Sucesso
Robson_20.jpg	Robson Silva 1111114	0.315979	1.3072	Sucesso
Robson_21.jpg	Robson Silva 1111114	0.414376	1.29783	Sucesso
Robson_23.jpg	Robson Silva 1111114	0.468793	1.25478	Sucesso
Robson_24.jpg	Robson Silva 1111114	0.361241	1.2628	Sucesso
Robson_25.jpg	Robson Silva 1111114	0.471257	1.29585	Sucesso
Robson_26.jpg	Robson Silva 1111114	0.402217	1.30354	Sucesso
Robson_28.jpg	Robson Silva 1111114	0.427212	1.29314	Sucesso
Robson_29.jpg	Robson Silva 1111114	0.489488	1.25113	Sucesso
Robson_31.jpg	Robson Silva 1111114	0.439782	1.30343	Sucesso
Robson_32.jpg	Robson Silva 1111114	0.451032	1.28678	Sucesso
Robson_33.jpg	Robson Silva 1111114	0.414376	1.26085	Sucesso
Robson_34.jpg	Robson Silva 1111114	0.42418	1.28359	Sucesso
Nune_1.jpg	Unknown	0.728553	0.892282	Falhou
Nune_2.jpg	Unknown	0.748399	0.830548	Falhou
Nune_3.jpg	Unknown	0.791909	0.891929	Falhou
Nune_4.jpg	Unknown	0.741905	0.844395	Falhou
Nune_5.jpg	Unknown	0.630136	0.889414	Falhou
Nune_6.jpg	Unknown	0.746968	1.29948	Falhou
Nune_7.jpg	Unknown	0.66439	1.06026	Falhou
Nune_8.jpg	Unknown	0.708895	0.858909	Falhou
Nune_9.jpg	Unknown	0.654761	0.844457	Falhou
Nune_10.jpg	Nune nascimento 1111112	0.54204	1.15581	Sucesso
Nune_11.jpg	Nune nascimento 1111112	0.478254	1.69095	Sucesso
Nune_12.jpg	Unknown	0.61162	0.888374	Falhou
Nune_13.jpg	Nune nascimento 1111112	0.553149	1.1675	Sucesso
Nune_14.jpg	Nune nascimento 1111112	0.524975	1.179	Sucesso
Nune_15.jpg	Nune nascimento 1111112	0.505763	1.19717	Sucesso
Nune_16.jpg	Nune nascimento 1111112	0.52663	1.18239	Sucesso
Nune_18.jpg	Unknown	0.728553	0.850142	Falhou
Nune_19.jpg	Unknown	0.791909	0.874217	Falhou
Nune_20.jpg	Unknown	0.726935	0.891125	Falhou
Nune_21.jpg	Nune nascimento 1111112	0.472558	1.18937	Sucesso
Nune_22.jpg	Nune nascimento 1111112	0.537245	1.2477	Sucesso
Nune_23.jpg	Unknown	0.689223	0.88012	Falhou
Nune_24.jpg	Unknown	0.804774	0.878547	Falhou
Nune_25.jpg	Unknown	0.720605	0.894152	Falhou
Nune_26.jpg	Unknown	0.624868	0.867992	Falhou
Nune_27.jpg	Unknown	0.752905	0.862899	Falhou
Nune_30.jpg	Unknown	0.61162	0.886562	Falhou
Nune_31.jpg	Unknown	0.728553	1.36496	Falhou
Nune_32.jpg	Unknown	0.66439	1.01731	Falhou
Nune_33.jpg	Unknown	0.630136	0.872986	Falhou
Nune_34.jpg	Unknown	0.791909	0.898622	Falhou

Lucas_0.jpg	Lucas Telles 1111111	0.419685	2.04474	Sucesso
Lucas_2.jpg	Lucas Telles 1111111	0.574719	1.3062	Sucesso
Lucas_3.jpg	Lucas Telles 1111111	0.53544	1.31216	Sucesso
Lucas_5.jpg	Lucas Telles 1111111	0.57558	1.3114	Sucesso
Lucas_6.jpg	Unknown	0.604901	1.00647	Falhou
Lucas_7.jpg	Lucas Telles 1111111	0.56795	1.29911	Sucesso
Lucas_8.jpg	Lucas Telles 1111111	0.583176	1.32518	Sucesso
Lucas_10.jpg	Lucas Telles 1111111	0.531642	1.27467	Sucesso
Lucas_11.jpg	Unknown	0.657624	1.46849	Falhou
Lucas_12.jpg	Lucas Telles 1111111	0.530283	1.3755	Sucesso
Lucas_14.jpg	Unknown	0.604388	0.936605	Falhou
Lucas_15.jpg	Lucas Telles 1111111	0.524545	1.67466	Sucesso
Lucas_17.jpg	Lucas Telles 1111111	0.372664	1.56634	Sucesso
Lucas_19.jpg	Lucas Telles 1111111	0.572806	1.28687	Sucesso
Lucas_20.jpg	Lucas Telles 1111111	0.389479	1.32755	Sucesso
Lucas_21.jpg	Lucas Telles 1111111	0.500076	1.31406	Sucesso
Lucas_22.jpg	Lucas Telles 1111111	0.498972	1.28015	Sucesso
Lucas_24.jpg	Lucas Telles 1111111	0.445803	1.31913	Sucesso
Lucas_25.jpg	Lucas Telles 1111111	0.445803	1.31021	Sucesso
Lucas_26.jpg	Lucas Telles 1111111	0.417575	1.36172	Sucesso
Lucas_28.jpg	Lucas Telles 1111111	0.439836	1.31648	Sucesso
Lucas_29.jpg	Unknown	0.657339	0.959976	Falhou
Lucas_30.jpg	Unknown	0.65803	0.972118	Falhou
Lucas_31.jpg	Lucas Telles 1111111	0.454663	1.28637	Sucesso
Lucas_32.jpg	Unknown	0.66109	1.0382	Falhou
Lucas_33.jpg	Lucas Telles 1111111	0.500158	1.22915	Sucesso
Lucas_34.jpg	Lucas Telles 1111111	0.42429	1.33117	Sucesso
Rafael_0.jpg	Unknown	0.690679	0.975606	Falhou
Rafael_1.jpg	Rafael Ribeiro 1111113	0.543035	1.3138	Sucesso
Rafael_2.jpg	Rafael Ribeiro 1111113	0.52483	1.32174	Sucesso
Rafael_4.jpg	Rafael Ribeiro 1111113	0.499806	1.2739	Sucesso
Rafael_5.jpg	Rafael Ribeiro 1111113	0.448446	1.29407	Sucesso
Rafael_6.jpg	Rafael Ribeiro 1111113	0.494571	1.31343	Sucesso
Rafael_7.jpg	Unknown	0.611468	0.96317	Falhou
Rafael_8.jpg	Rafael Ribeiro 1111113	0.535734	1.33042	Sucesso
Rafael_10.jpg	Rafael Ribeiro 1111113	0.546249	1.28596	Sucesso
Rafael_11.jpg	Rafael Ribeiro 1111113	0.499559	1.3009	Sucesso
Rafael_12.jpg	Rafael Ribeiro 1111113	0.547781	1.30924	Sucesso
Rafael_13.jpg	Rafael Ribeiro 1111113	0.532825	1.27662	Sucesso
Rafael_14.jpg	Unknown	0.538901	1.45985	Sucesso
Rafael_16.jpg	Rafael Ribeiro 1111113	0.502877	1.27186	Sucesso
Rafael_17.jpg	Rafael Ribeiro 1111113	0.555504	1.29393	Sucesso
Rafael_18.jpg	Rafael Ribeiro 1111113	0.531744	1.30073	Sucesso
Rafael_19.jpg	Unknown	0.606704	0.963208	Falhou
Rafael_20.jpg	Unknown	0.648147	0.968657	Falhou
Rafael_21.jpg	Rafael Ribeiro 1111113	0.554632	1.30799	Sucesso
Rafael_23.jpg	Rafael Ribeiro 1111113	0.564083	1.32113	Sucesso
Rafael_24.jpg	Rafael Ribeiro 1111113	0.561799	1.26183	Sucesso
Rafael_25.jpg	Unknown	0.67084	0.975591	Falhou
Rafael_26.jpg	Unknown	0.632511	0.967636	Falhou
Rafael_27.jpg	Rafael Ribeiro 1111113	0.572162	1.28794	Sucesso
Rafael_28.jpg	Unknown	0.625572	1.0023	Falhou
Rafael_29.jpg	Unknown	0.642632	0.981753	Falhou
Rafael_30.jpg	Rafael Ribeiro 1111113	0.573085	1.29576	Sucesso
Rafael_31.jpg	Unknown	0.648147	0.96727	Falhou
Rafael_3				

Japa_2.jpg	Japa Kim 1622211	0.443753	1.95325	Sucesso
Japa_4.jpg	Japa Kim 1622211	0.381367	1.31301	Sucesso
Japa_5.jpg	Japa Kim 1622211	0.402206	1.31129	Sucesso
Japa_7.jpg	Japa Kim 1622211	0.367248	1.27004	Sucesso
Japa_8.jpg	Japa Kim 1622211	0.350962	1.48465	Sucesso
Japa_10.jpg	Japa Kim 1622211	0.379294	1.28049	Sucesso
Japa_11.jpg	Japa Kim 1622211	0.438586	1.23153	Sucesso
Japa_14.jpg	Japa Kim 1622211	0.37006	1.27349	Sucesso
Japa_15.jpg	Japa Kim 1622211	0.392447	1.28387	Sucesso
Japa_16.jpg	Japa Kim 1622211	0.411494	1.27818	Sucesso
Japa_17.jpg	Japa Kim 1622211	0.421273	1.30346	Sucesso
Japa_19.jpg	Japa Kim 1622211	0.427177	1.24846	Sucesso
Japa_20.jpg	Japa Kim 1622211	0.449316	1.25777	Sucesso
Japa_21.jpg	Japa Kim 1622211	0.554768	1.17862	Sucesso
Japa_22.jpg	Japa Kim 1622211	0.416085	2.11123	Sucesso
Japa_24.jpg	Japa Kim 1622211	0.375422	1.29691	Sucesso
Japa_25.jpg	Japa Kim 1622211	0.392222	1.31128	Sucesso
Japa_27.jpg	Japa Kim 1622211	0.385731	1.22446	Sucesso
Japa_28.jpg	Japa Kim 1622211	0.370145	1.2391	Sucesso
Japa_29.jpg	Japa Kim 1622211	0.35557	1.30575	Sucesso
Japa_30.jpg	Japa Kim 1622211	0.313288	1.29652	Sucesso
Japa_32.jpg	Japa Kim 1622211	0.383826	1.89362	Sucesso
JapaMan_0.jpg	JapaMan KimK 1234567	0.502933	1.45714	Sucesso
JapaMan_2.jpg	JapaMan KimK 1234567	0.596395	1.28601	Sucesso
JapaMan_3.jpg	JapaMan KimK 1234567	0.410742	1.35935	Sucesso
JapaMan_4.jpg	JapaMan KimK 1234567	0.530969	1.29688	Sucesso
JapaMan_8.jpg	JapaMan KimK 1234567	0.56011	1.32261	Sucesso
JapaMan_9.jpg	JapaMan KimK 1234567	0.460632	1.33623	Sucesso
JapaMan_10.jpg	JapaMan KimK 1234567	0.432079	1.31512	Sucesso
JapaMan_11.jpg	JapaMan KimK 1234567	0.433071	1.31207	Sucesso
JapaMan_13.jpg	JapaMan KimK 1234567	0.418638	1.31266	Sucesso
JapaMan_14.jpg	JapaMan KimK 1234567	0.547653	1.28847	Sucesso
JapaMan_15.jpg	JapaMan KimK 1234567	0.556205	1.32843	Sucesso
JapaMan_18.jpg	JapaMan KimK 1234567	0.463739	1.48733	Sucesso
JapaMan_19.jpg	JapaMan KimK 1234567	0.487505	1.34526	Sucesso
JapaMan_21.jpg	JapaMan KimK 1234567	0.439676	2.03538	Sucesso
JapaMan_23.jpg	JapaMan KimK 1234567	0.552039	1.25553	Sucesso
JapaMan_24.jpg	JapaMan KimK 1234567	0.511565	1.29159	Sucesso
JapaMan_25.jpg	JapaMan KimK 1234567	0.497703	1.28869	Sucesso
JapaMan_29.jpg	JapaMan KimK 1234567	0.378683	1.43097	Sucesso
JapaMan_30.jpg	JapaMan KimK 1234567	0.450354	1.29771	Sucesso
JapaMan_31.jpg	JapaMan KimK 1234567	0.318502	1.35167	Sucesso
JapaChild_0.jpg	JapaChild Assaki 14132266	0.582624	1.20084	Sucesso
JapaChild_1.jpg	JapaChild Assaki 14132266	0.568559	1.21613	Sucesso
JapaChild_2.jpg	JapaChild Assaki 14132266	0.519605	1.15887	Sucesso
JapaChild_3.jpg	JapaChild Assaki 14132266	0.582624	1.24377	Sucesso
JapaChild_4.jpg	JapaChild Assaki 14132266	0.244449	1.16347	Sucesso
JapaChild_5.jpg	JapaChild Assaki 14132266	0.601061	0.912433	Falhou
JapaChild_6.jpg	JapaChild Assaki 14132266	0.582624	1.22481	Sucesso
JapaChild_7.jpg	JapaChild Assaki 14132266	0.642181	0.926148	Falhou
JapaChild_8.jpg	JapaChild Assaki 14132266	0.59337	1.91317	Sucesso
JapaChild_10.jpg	JapaChild Assaki 14132266	0.551592	1.18502	Sucesso
JapaChild_11.jpg	JapaChild Assaki 14132266	0.551592	1.18829	Sucesso
JapaChild_12.jpg	JapaChild Assaki 14132266	0.36469	1.15533	Sucesso
JapaChild_13.jpg	JapaChild Assaki 14132266	0.583093	1.2521	Sucesso
JapaChild_15.jpg	Unknown	0.627366	0.937847	Falhou
JapaChild_16.jpg	JapaChild Assaki 14132266	0.526704	1.21092	Sucesso
JapaChild_17.jpg	JapaChild Assaki 14132266	0.564543	1.22088	Sucesso
JapaChild_19.jpg	JapaChild Assaki 14132266	0.567856	1.83961	Sucesso
JapaChild_20.jpg	JapaChild Assaki 14132266	0.468304	1.13321	Sucesso
JapaChild_23.jpg	JapaChild Assaki 14132266	0.573665	1.2375	Sucesso
JapaChild_24.jpg	JapaChild Assaki 14132266	0.582624	1.22565	Sucesso
JapaChild_25.jpg	JapaChild Assaki 14132266	0.568559	1.2385	Sucesso
JapaChild_26.jpg	JapaChild Assaki 14132266	0.583093	1.24247	Sucesso
JapaChild_28.jpg	Unknown	0.642181	0.913386	Falhou
JapaChild_29.jpg	JapaChild Assaki 14132266	0.524968	2.38463	Sucesso
JapaChild_31.jpg	JapaChild Assaki 14132266	0.514311	1.1905	Sucesso
JapaChild_32.jpg	JapaChild Assaki 14132266	0.537096	1.23942	Sucesso
JapaChild_34.jpg	JapaChild Assaki 14132266	0.582624	1.23735	Sucesso
JapaNew_0.jpg	Unknown	0.667111	0.883515	Falhou
JapaNew_1.jpg	Unknown	0.643172	1.07433	Falhou
JapaNew_2.jpg	Unknown	0.650356	0.891729	Falhou
JapaNew_3.jpg	Unknown	0.630966	0.827799	Falhou
JapaNew_4.jpg	Unknown	0.630966	1.35125	Falhou
JapaNew_5.jpg	JapaNew Assaki 1223226	0.51853	1.30305	Sucesso
JapaNew_7.jpg	JapaNew Assaki 1223226	0.581337	1.22713	Sucesso
JapaNew_8.jpg	JapaNew Assaki 1223226	0.581337	1.22493	Sucesso
JapaNew_9.jpg	JapaNew Assaki 1223226	0.556737	1.23716	Sucesso
JapaNew_10.jpg	Unknown	0.636605	0.906016	Falhou
JapaNew_11.jpg	Unknown	0.667111	0.881223	Falhou
JapaNew_12.jpg	Unknown	0.673094	0.856932	Falhou
JapaNew_13.jpg	Unknown	0.639006	0.855612	Falhou
JapaNew_15.jpg	JapaNew Assaki 1223226	0.522185	1.81312	Sucesso
JapaNew_17.jpg	JapaNew Assaki 1223226	0.563336	1.22084	Sucesso
JapaNew_18.jpg	JapaNew Assaki 1223226	0.563366	1.22059	Sucesso
JapaNew_19.jpg	JapaNew Assaki 1223226	0.598778	1.21929	Sucesso
JapaNew_20.jpg	JapaNew Assaki 1223226	0.598778	1.22326	Sucesso
JapaNew_21.jpg	Unknown	0.637358	0.920764	Falhou
JapaNew_22.jpg	Unknown	0.624964	0.887877	Falhou
JapaNew_23.jpg	Unknown	0.642381	0.843845	Falhou
JapaNew_24.jpg	Unknown	0.630207	0.866971	Falhou
JapaNew_25.jpg	JapaNew Assaki 1223226	0.459535	1.81957	Sucesso
JapaNew_27.jpg	Unknown	0.641934	0.945901	Falhou
JapaNew_28.jpg	Unknown	0.629369	0.906993	Falhou
JapaNew_29.jpg	Unknown	0.629369	0.90863	Falhou
JapaNew_30.jpg	JapaNew Assaki 1223226	0.522098	1.74926	Sucesso
JapaNew_32.jpg	Unknown	0.650356	0.880847	Falhou
JapaNew_33.jpg	Unknown	0.643172	1.05235	Falhou
JapaNew_34.jpg	JapaNew Assaki 1223226	0.545831	1.24493	Sucesso
JapaFat_0.jpg	Unknown	0.602244	0.943346	Sucesso
JapaFat_1.jpg	Unknown	0.602244	0.938754	Sucesso
JapaFat_2.jpg	Unknown	0.602244	0.937826	Sucesso
JapaFat_3.jpg	JapaChild Assaki 14132266	0.561051	1.25827	Falhou
JapaFat_6.jpg	JapaWhite Houriish 2963226	0.564662	1.42744	Falhou
JapaFat_8.jpg	JapaWhite Houriish 2963226	0.564662	1.24398	Falhou

JapaFat_9.jpg	JapaChild Assaki 14132266	0.566324	1.22655	Falhou
JapaFat_10.jpg	JapaChild Assaki 14132266	0.57888	1.2467	Falhou
JapaFat_11.jpg	JapaChild Assaki 14132266	0.57888	1.25009	Falhou
JapaFat_12.jpg	JapaNew Assaki 1223226	0.493614	1.25823	Falhou
JapaFat_14.jpg	JapaNew Assaki 1223226	0.481384	1.23851	Falhou
JapaFat_15.jpg	JapaNew Assaki 1223226	0.481384	1.24862	Falhou
JapaFat_16.jpg	JapaNew Assaki 1223226	0.481384	1.21462	Falhou
JapaFat_17.jpg	Unknown	0.645136	0.921146	Sucesso
JapaFat_18.jpg	Unknown	0.547243	1.23421	Falhou
JapaFat_19.jpg	JapaChild Assaki 14132266	0.594799	1.25897	Falhou
JapaFat_21.jpg	JapaChild Assaki 14132266	0.56793	2.01243	Falhou
JapaFat_22.jpg	JapaChild Assaki 14132266	0.56793	1.24249	Falhou
JapaFat_23.jpg	JapaChild Assaki 14132266	0.570267	1.26339	Falhou
JapaFat_25.jpg	JapaNew Assaki 1223226	0.458574	1.23271	Falhou
JapaFat_26.jpg	JapaNew Assaki 1223226	0.458574	1.22091	Falhou
JapaFat_27.jpg	JapaNew Assaki 1223226	0.458574	1.21758	Falhou
JapaFat_28.jpg	JapaNew Assaki 1223226	0.553982	1.18367	Falhou
JapaFat_29.jpg	Unknown	0.615522	0.942581	Sucesso
JapaFat_30.jpg	Unknown	0.615522	0.941979	Sucesso
JapaFat_31.jpg	JapaChild Assaki 14132266	0.554089	1.42086	Falhou
JapaFat_33.jpg	Unknown	0.62542	0.915031	Sucesso
JapaFat_34.jpg	Unknown	0.62542	0.936242	Sucesso
JapaUS_0.jpg	Unknown	0.678534	0.99462	Falhou
JapaUS_1.jpg	Unknown	0.707943	1.59683	Falhou
JapaUS_3.jpg	Unknown	0.656741	1.16057	Falhou
JapaUS_4.jpg	Unknown	0.655285	0.979195	Falhou
JapaUS_5.jpg	Unknown	0.655285	0.979913	Falhou
JapaUS_6.jpg	Unknown	0.617153	0.954312	Falhou
JapaUS_7.jpg	Unknown	0.625902	1.37302	Falhou
JapaUS_8.jpg	Unknown	0.644889	1.1168	Falhou
JapaUS_9.jpg	JapaUS Nashira 4963226	0.598045	1.30413	Sucesso
JapaUS_10.jpg	Unknown	0.692207	0.955953	Falhou
JapaUS_11.jpg	Unknown	0.712927	0.943289	Falhou
JapaUS_12.jpg	Unknown	0.635101	0.971611	Falhou
JapaUS_13.jpg	Unknown	0.659283	0.935541	Falhou
JapaUS_14.jpg	Unknown	0.656294	0.996934	Falhou
JapaUS_15.jpg	Unknown	0.66669	1.00596	Falhou
JapaUS_16.jpg	Unknown	0.612936	1.02391	Falhou
JapaUS_17.jpg	Unknown	0.69603	1.56417	Falhou
JapaUS_19.jpg	Unknown	0.650085	1.29182	Falhou
JapaUS_20.jpg	Unknown	0.697112	0.958814	Falhou
JapaUS_21.jpg	Unknown	0.706704	0.988669	Falhou
JapaUS_22.jpg	Unknown	0.71176	0.980444	Falhou
JapaUS_23.jpg	Unknown	0.701844	1.0065	Falhou
JapaUS_24.jpg	Unknown	0.60803	0.991039	Falhou
JapaUS_25.jpg	Unknown	0.642892	0.945672	Falhou
JapaUS_26.jpg	Unknown	0.605552	1.01093	Falhou
JapaUS_27.jpg	Unknown	0.707943	1.60581	Falhou
JapaUS_29.jpg	Unknown	0.678534	1.176	Falhou
JapaUS_30.jpg	Unknown	0.612936	1.00156	Falhou
JapaUS_32.jpg	Unknown	0.692207	0.956983	Falhou
JapaUS_33.jpg	Unknown	0.617153	1.28725	Falhou
JapaUS_34.jpg	Unknown	0.71176	1.10346	Falhou
JapaCN_0.jpg	JapaCN Nawan 7963226	0.462851	1.28816	Sucesso
JapaCN_1.jpg	JapaCN Nawan 7963226	0.58745	1.27749	Sucesso
JapaCN_2.jpg	JapaCN Nawan 7963226	0.491287	1.28959	Sucesso
JapaCN_3.jpg	JapaCN Nawan 7963226	0.591459	1.2752	Sucesso
JapaCN_5.jpg	JapaCN Nawan 7963226	0.526742	1.26885	Sucesso
JapaCN_6.jpg	JapaCN Nawan 7963226	0.468455	1.27693	Sucesso
JapaCN_7.jpg	JapaCN Nawan 7963226	0.546453	1.31547	Sucesso
JapaCN_8.jpg	JapaCN Nawan 7963226	0.570348	1.29464	Sucesso
JapaCN_10.jpg	Unknown	0.61298	0.95796	Falhou
JapaCN_11.jpg	Unknown	0.663061	0.973322	Falhou
JapaCN_12.jpg	Unknown	0.67156	0.98917	Falhou
JapaCN_13.jpg	Unknown	0.467957	1.96196	Sucesso
JapaCN_14.jpg	Unknown	0.491287	1.25256	Sucesso
JapaCN_15.jpg	Unknown	0.626222	0.948028	Falhou
JapaCN_16.jpg	Unknown	0.633809	0.97962	Falhou
JapaCN_17.jpg	Unknown	0.659816	1.01159	Falhou
JapaCN_18.jpg	JapaCN Nawan 7963226	0.475576	1.29149	Sucesso
JapaCN_19.jpg	Unknown	0.680918	0.960445	Falhou
JapaCN_20.jpg	Unknown	0.674688	0.958339	Falhou
JapaCN_21.jpg	Unknown	0.683943	0.960618	Falhou
JapaCN_22.jpg	Unknown	0.725732	1.50478</td	

JapaWhite_27.jpg	JapaWhite Hourish 2963226	0.313192	1.46524	Sucesso
JapaWhite_29.jpg	JapaWhite Hourish 2963226	0.484005	1.28095	Sucesso
JapaWhite_30.jpg	JapaWhite Hourish 2963226	0.400579	1.27799	Sucesso
JapaWhite_31.jpg	JapaWhite Hourish 2963226	0.472319	1.28376	Sucesso
JapaWhite_32.jpg	JapaWhite Hourish 2963226	0.248195	1.28209	Sucesso
JapaWhite_34.jpg	JapaWhite Hourish 2963226	0.248195	1.2742	Sucesso
JapaOld_0.jpg	JapaOld Akira 1763226	0.529076	1.40393	Sucesso
JapaOld_1.jpg	JapaOld Akira 1763226	0.490306	1.31742	Sucesso
JapaOld_2.jpg	JapaOld Akira 1763226	0.532657	1.31418	Sucesso
JapaOld_4.jpg	JapaOld Akira 1763226	0.493578	1.34303	Sucesso
JapaOld_5.jpg	JapaOld Akira 1763226	0.397407	1.27545	Sucesso
JapaOld_6.jpg	JapaOld Akira 1763226	0.532657	1.34411	Sucesso
JapaOld_8.jpg	JapaOld Akira 1763226	0.486256	1.36374	Sucesso
JapaOld_9.jpg	JapaOld Akira 1763226	0.51115	1.36463	Sucesso
JapaOld_10.jpg	JapaOld Akira 1763226	0.511844	1.35665	Sucesso
JapaOld_12.jpg	JapaOld Akira 1763226	0.494006	1.3368	Sucesso
JapaOld_13.jpg	JapaOld Akira 1763226	0.578758	1.28005	Sucesso
JapaOld_14.jpg	JapaOld Akira 1763226	0.519614	1.35263	Sucesso
JapaOld_15.jpg	JapaOld Akira 1763226	0.529076	1.43197	Sucesso
JapaOld_17.jpg	JapaOld Akira 1763226	0.492271	1.3295	Sucesso
JapaOld_18.jpg	JapaOld Akira 1763226	0.485922	1.37126	Sucesso
JapaOld_19.jpg	JapaOld Akira 1763226	0.511546	1.33562	Sucesso
JapaOld_21.jpg	JapaOld Akira 1763226	0.561856	1.29405	Sucesso
JapaOld_22.jpg	JapaOld Akira 1763226	0.561856	1.29453	Sucesso
JapaOld_23.jpg	JapaOld Akira 1763226	0.521134	1.38275	Sucesso
JapaOld_25.jpg	JapaOld Akira 1763226	0.590473	1.35428	Sucesso
JapaOld_27.jpg	JapaOld Akira 1763226	0.566533	1.37561	Sucesso
JapaOld_28.jpg	JapaOld Akira 1763226	0.535629	1.38288	Sucesso
JapaOld_30.jpg	JapaOld Akira 1763226	0.529076	2.03161	Sucesso
JapaOld_32.jpg	JapaOld Akira 1763226	0.590473	1.32461	Sucesso
JapaOld_33.jpg	JapaOld Akira 1763226	0.521134	1.36354	Sucesso
JapaOld_34.jpg	JapaOld Akira 1763226	0.519614	1.39342	Sucesso
JapaShort_1.jpg	JapaShort Okuri 1232266	0.422615	1.30417	Sucesso
JapaShort_2.jpg	JapaShort Okuri 1232266	0.421986	1.29675	Sucesso
JapaShort_3.jpg	JapaShort Okuri 1232266	0.460323	1.2667	Sucesso
JapaShort_4.jpg	JapaShort Okuri 1232266	0.511546	1.91147	Sucesso
JapaShort_6.jpg	JapaShort Okuri 1232266	0.424969	1.35592	Sucesso
JapaShort_7.jpg	JapaShort Okuri 1232266	0.424969	1.27604	Sucesso
JapaShort_8.jpg	JapaShort Okuri 1232266	0.352795	1.33257	Sucesso
JapaShort_10.jpg	JapaShort Okuri 1232266	0.352501	1.31335	Sucesso
JapaShort_11.jpg	JapaShort Okuri 1232266	0.347619	1.28753	Sucesso
JapaShort_12.jpg	JapaShort Okuri 1232266	0.412218	1.28344	Sucesso
JapaShort_13.jpg	JapaShort Okuri 1232266	0.370466	1.27239	Sucesso
JapaShort_15.jpg	JapaShort Okuri 1232266	0.369614	2.11369	Sucesso
JapaShort_16.jpg	JapaShort Okuri 1232266	0.379628	1.27804	Sucesso
JapaShort_17.jpg	JapaShort Okuri 1232266	0.337615	1.32891	Sucesso
JapaShort_19.jpg	JapaShort Okuri 1232266	0.353686	1.31274	Sucesso
JapaShort_20.jpg	JapaShort Okuri 1232266	0.384501	1.27444	Sucesso
JapaShort_21.jpg	JapaShort Okuri 1232266	0.544987	1.26659	Sucesso
JapaShort_22.jpg	JapaShort Okuri 1232266	0.589048	1.28184	Sucesso
JapaShort_24.jpg	JapaShort Okuri 1232266	0.323678	1.95137	Sucesso
JapaShort_26.jpg	JapaShort Okuri 1232266	0.501975	1.30486	Sucesso
JapaShort_27.jpg	JapaShort Okuri 1232266	0.509044	1.25052	Sucesso
JapaShort_28.jpg	JapaShort Okuri 1232266	0.397541	1.30911	Sucesso
JapaShort_29.jpg	JapaShort Okuri 1232266	0.352795	1.34544	Sucesso

JapaShort_31.jpg	JapaShort Okuri 1232266	0.443096	1.30671	Sucesso
JapaHair_0.jpg	JapaWhite Hourish 2963226	0.511844	1.24285	Falhou
JapaHair_1.jpg	Unknown	0.626232	0.928853	Sucesso
JapaHair_3.jpg	JapaUS Nashira 4963226	0.583009	1.85155	Falhou
JapaHair_4.jpg	JapaUS Nashira 4963226	0.574655	1.29109	Falhou
JapaHair_5.jpg	JapaShort Okuri 1232266	0.508948	1.30418	Falhou
JapaHair_7.jpg	JapaUS Nashira 4963226	0.528616	1.23841	Falhou
JapaHair_8.jpg	JapaWhite Hourish 2963226	0.423955	1.2481	Falhou
JapaHair_9.jpg	JapaWhite Hourish 2963226	0.433921	1.23425	Falhou
JapaHair_10.jpg	JapaWhite Hourish 2963226	0.537979	1.22514	Falhou
JapaHair_11.jpg	Unknown	0.654008	0.941186	Sucesso
JapaHair_12.jpg	Unknown	0.632239	0.942875	Sucesso
JapaHair_13.jpg	Unknown	0.623511	0.979892	Sucesso
JapaHair_14.jpg	Unknown	0.602046	0.965437	Sucesso
JapaHair_15.jpg	Unknown	0.609358	0.948639	Sucesso
JapaHair_16.jpg	JapaWhite Hourish 2963226	0.550213	1.2686	Falhou
JapaHair_17.jpg	Unknown	0.641339	0.943842	Sucesso
JapaHair_18.jpg	JapaWhite Hourish 2963226	0.495517	1.88125	Falhou
JapaHair_20.jpg	Unknown	0.609417	0.898374	Sucesso
JapaHair_23.jpg	JapaShort Okuri 1232266	0.355387	1.24624	Falhou
JapaHair_24.jpg	JapaWhite Hourish 2963226	0.462563	1.61263	Falhou
JapaHair_25.jpg	JapaWhite Hourish 2963226	0.488596	1.25513	Falhou
JapaHair_26.jpg	JapaFun Nagawa 3963226	0.521169	1.27605	Falhou
JapaHair_28.jpg	Unknown	0.657668	0.985354	Sucesso
JapaLong_0.jpg	Unknown	0.735994	0.97171	Falhou
JapaLong_1.jpg	Unknown	0.754843	0.978478	Falhou
JapaLong_2.jpg	Unknown	0.72343	1.00683	Falhou
JapaLong_3.jpg	JapaLong Oishi 1232257	0.586901	1.31326	Sucesso
JapaLong_6.jpg	Unknown	0.771221	1.21536	Falhou
JapaLong_7.jpg	Unknown	0.701866	0.984291	Falhou
JapaLong_8.jpg	Unknown	0.782755	1.00044	Falhou
JapaLong_9.jpg	Unknown	0.828582	1.00205	Falhou
JapaLong_10.jpg	Unknown	0.75865	1.06666	Falhou
JapaLong_11.jpg	JapaLong Oishi 1232257	0.560491	1.29116	Sucesso
JapaLong_12.jpg	Unknown	0.68782	0.898684	Falhou
JapaLong_13.jpg	JapaLong Oishi 1232257	0.548809	1.2764	Sucesso
JapaLong_14.jpg	JapaLong Oishi 1232257	0.564448	1.30665	Sucesso
JapaLong_15.jpg	JapaLong Oishi 1232257	0.569404	1.28756	Sucesso
JapaLong_16.jpg	JapaLong Oishi 1232257	0.528762	1.29487	Sucesso
JapaLong_21.jpg	Unknown	0.763377	1.05288	Falhou
JapaLong_22.jpg	Unknown	0.834504	0.941789	Falhou
JapaLong_23.jpg	Unknown	0.75984	0.977462	Falhou
JapaLong_24.jpg	Unknown	0.759223	0.913274	Falhou
JapaLong_26.jpg	Unknown	0.640719	0.983551	Falhou
JapaLong_27.jpg	Unknown	0.652067	0.941895	Falhou
JapaLong_28.jpg	Unknown	0.771221	0.994188	Falhou
JapaLong_29.jpg	JapaLong Oishi 1232257	0.569404	1.23669	Sucesso
JapaLong_30.jpg	JapaLong Oishi 1232257	0.569404	1.25425	Sucesso
JapaLong_31.jpg	Unknown	0.763377	0.985056	Falhou
JapaLong_32.jpg	Unknown	0.68782	0.886554	Falhou

Média	0.54	1.22
Desvio Padrão	0.11	0.24
Total Amostras	434	
Falhou	37.1%	
Sucesso	62.9%	

Tabela 3: Threshold = 0.7

Arquivo de Entrada	Nome Identificado	Similaridade	Tempo(s)	Resultado
Robson_0.jpg	Robson Silva 1111114	0.363329	1.35191	Sucesso
Robson_1.jpg	Robson Silva 1111114	0.374507	1.28119	Sucesso
Robson_2.jpg	Robson Silva 1111114	0.380971	1.2866	Sucesso
Robson_4.jpg	Robson Silva 1111114	0.38645	1.23752	Sucesso
Robson_5.jpg	Robson Silva 1111114	0.398453	1.28245	Sucesso
Robson_6.jpg	Robson Silva 1111114	0.364721	1.48909	Sucesso
Robson_7.jpg	Robson Silva 1111114	0.33219	1.29312	Sucesso
Robson_9.jpg	Robson Silva 1111114	0.429284	1.25065	Sucesso
Robson_10.jpg	Robson Silva 1111114	0.467387	1.27755	Sucesso
Robson_11.jpg	Robson Silva 1111114	0.325547	1.28944	Sucesso
Robson_12.jpg	Robson Silva 1111114	0.483048	1.27798	Sucesso
Robson_14.jpg	Robson Silva 1111114	0.342467	1.28926	Sucesso
Robson_15.jpg	Robson Silva 1111114	0.382108	1.29305	Sucesso
Robson_16.jpg	Robson Silva 1111114	0.302352	1.30751	Sucesso
Robson_18.jpg	Robson Silva 1111114	0.464708	1.21196	Sucesso
Robson_19.jpg	Robson Silva 1111114	0.530395	1.26509	Sucesso
Robson_20.jpg	Robson Silva 1111114	0.315979	1.30035	Sucesso
Robson_21.jpg	Robson Silva 1111114	0.414376	1.27982	Sucesso
Robson_22.jpg	Robson Silva 1111114	0.424118	1.29434	Sucesso
Robson_24.jpg	Robson Silva 1111114	0.361241	1.25983	Sucesso
Robson_25.jpg	Robson Silva 1111114	0.471257	1.28462	Sucesso
Robson_26.jpg	Robson Silva 1111114	0.402217	1.27066	Sucesso
Robson_27.jpg	Robson Silva 1111114	0.343784	1.31596	Sucesso
Robson_29.jpg	Robson Silva 1111114	0.489488	1.24474	Sucesso
Robson_31.jpg	Robson Silva 1111114	0.439782	1.2967	Sucesso
Robson_32.jpg	Robson Silva 1111114	0.451032	1.2886	Sucesso
Robson_33.jpg	Robson Silva 1111114	0.414376	1.25729	Sucesso
Robson_34.jpg	Robson Silva 1111114	0.424118	1.2795	Sucesso
Nune_1.jpg	Unknown	0.728553	0.865973	Falhou
Nune_2.jpg	Unknown	0.748399	0.850675	Falhou
Nune_3.jpg	Unknown	0.791909	0.890659	Falhou
Nune_4.jpg	Unknown	0.741905	0.849784	Falhou
Nune_5.jpg	Nune nascimento 1111112	0.630136	1.19812	Sucesso
Nune_6.jpg	Unknown	0.746968	1.0214	Falhou
Nune_7.jpg	Nune nascimento 1111112	0.66439	1.2172	Sucesso
Nune_8.jpg	Unknown	0.708895	0.86062	Falhou

Nune_9.jpg	Nune nascimento 1111112	0.654761	1.14943	Sucesso
Nune_10.jpg	Nune nascimento 1111112	0.54204	1.1518	Sucesso
Nune_11.jpg	Nune nascimento 1111112	0.478254	1.85557	Sucesso
Nune_13.jpg	Nune nascimento 1111112	0.553149	1.16163	Sucesso
Nune_14.jpg	Nune nascimento 1111112	0.524975	1.17686	Sucesso
Nune_15.jpg	Nune nascimento 1111112	0.505763	1.19958	Sucesso
Nune_17.jpg	Nune nascimento 1111112	0.531903	1.26786	Sucesso
Nune_18.jpg	Unknown	0.728553	0.868457	Falhou
Nune_19.jpg	Unknown	0.791909	0.859426	Falhou
Nune_20.jpg	Unknown	0.726935	0.863831	Falhou
Nune_21.jpg	Nune nascimento 1111112	0.472558	1.86744	Sucesso
Nune_23.jpg	Nune nascimento 1111112	0.689223	1.4097	Sucesso
Nune_24.jpg	Unknown	0.804774	0.875317	Falhou
Nune_25.jpg	Unknown	0.720605	0.883701	Falhou
Nune_26.jpg	Nune nascimento 1111112	0.624868	1.19635	Sucesso
Nune_27.jpg	Unknown	0.752905	0.863464	Falhou
Nune_30.jpg	Nune nascimento 1111112	0.61162	1.19289	Sucesso
Nune_31.jpg	Unknown	0.728553	1.36235	Falhou
Nune_32.jpg	Nune nascimento 1111112	0.664349	1.34368	Sucesso
Nune_34.jpg	Unknown	0.791909	0.899362	Falhou
Lucas_0.jpg	Lucas Telles 1111111	0.419685	1.29	

Lucas_22.jpg	Lucas Telles 1111111	0.498972	1.23814	Sucesso
Lucas_23.jpg	Lucas Telles 1111111	0.482112	1.30849	Sucesso
Lucas_24.jpg	Lucas Telles 1111111	0.445803	1.32083	Sucesso
Lucas_26.jpg	Lucas Telles 1111111	0.417575	1.32747	Sucesso
Lucas_27.jpg	Lucas Telles 1111111	0.583096	1.28568	Sucesso
Lucas_28.jpg	Lucas Telles 1111111	0.439836	1.31371	Sucesso
Lucas_29.jpg	Lucas Telles 1111111	0.657339	1.28202	Sucesso
Lucas_31.jpg	Lucas Telles 1111111	0.454663	1.28347	Sucesso
Lucas_32.jpg	Lucas Telles 1111111	0.661099	1.28406	Sucesso
Lucas_33.jpg	Lucas Telles 1111111	0.500158	1.20963	Sucesso
Lucas_34.jpg	Lucas Telles 1111111	0.42429	1.45404	Sucesso
Rafael_1.jpg	Rafael Ribeiro 1111113	0.543035	1.29349	Sucesso
Rafael_2.jpg	Rafael Ribeiro 1111113	0.52483	1.29989	Sucesso
Rafael_3.jpg	Rafael Ribeiro 1111113	0.62056	1.29338	Sucesso
Rafael_5.jpg	Rafael Ribeiro 1111113	0.448446	1.3013	Sucesso
Rafael_6.jpg	Rafael Ribeiro 1111113	0.494571	1.29467	Sucesso
Rafael_7.jpg	Rafael Ribeiro 1111113	0.611468	1.26773	Sucesso
Rafael_8.jpg	Rafael Ribeiro 1111113	0.535734	1.31074	Sucesso
Rafael_10.jpg	Rafael Ribeiro 1111113	0.546249	1.2788	Sucesso
Rafael_11.jpg	Rafael Ribeiro 1111113	0.499559	1.29352	Sucesso
Rafael_12.jpg	Rafael Ribeiro 1111113	0.547781	1.30727	Sucesso
Rafael_13.jpg	Rafael Ribeiro 1111113	0.532825	1.30782	Sucesso
Rafael_15.jpg	Rafael Ribeiro 1111113	0.504379	1.26417	Sucesso
Rafael_16.jpg	Rafael Ribeiro 1111113	0.502877	1.28618	Sucesso
Rafael_17.jpg	Rafael Ribeiro 1111113	0.555504	1.30741	Sucesso
Rafael_18.jpg	Rafael Ribeiro 1111113	0.531744	1.31769	Sucesso
Rafael_20.jpg	Rafael Ribeiro 1111113	0.648147	1.27435	Sucesso
Rafael_21.jpg	Rafael Ribeiro 1111113	0.554632	1.29159	Sucesso
Rafael_22.jpg	Rafael Ribeiro 1111113	0.614761	1.29576	Sucesso
Rafael_23.jpg	Rafael Ribeiro 1111113	0.564083	1.30735	Sucesso
Rafael_25.jpg	Rafael Ribeiro 1111113	0.67084	1.29403	Sucesso
Rafael_26.jpg	Rafael Ribeiro 1111113	0.632511	1.26976	Sucesso
Rafael_27.jpg	Rafael Ribeiro 1111113	0.572162	1.2882	Sucesso
Rafael_29.jpg	Rafael Ribeiro 1111113	0.642632	1.28737	Sucesso
Rafael_30.jpg	Rafael Ribeiro 1111113	0.573085	1.28857	Sucesso
Rafael_31.jpg	Rafael Ribeiro 1111113	0.648147	1.27637	Sucesso
Rafael_32.jpg	Rafael Ribeiro 1111113	0.566457	1.30947	Sucesso
Rafael_34.jpg	Rafael Ribeiro 1111113	0.62056	1.27444	Sucesso
Japa_0.jpg	Japa Kim 1622211	0.346483	1.24174	Sucesso
Japa_1.jpg	Japa Kim 1622211	0.382731	1.2567	Sucesso
Japa_2.jpg	Japa Kim 1622211	0.443753	1.2233	Sucesso
Japa_3.jpg	Japa Kim 1622211	0.288107	1.27942	Sucesso
Japa_5.jpg	Japa Kim 1622211	0.402206	1.28864	Sucesso
Japa_6.jpg	Japa Kim 1622211	0.352834	1.26271	Sucesso
Japa_7.jpg	Japa Kim 1622211	0.367248	1.2827	Sucesso
Japa_8.jpg	Japa Kim 1622211	0.350962	1.26835	Sucesso
Japa_10.jpg	Japa Kim 1622211	0.379294	1.26458	Sucesso
Japa_11.jpg	Japa Kim 1622211	0.438586	1.20336	Sucesso
Japa_14.jpg	Japa Kim 1622211	0.37005	1.2809	Sucesso
Japa_15.jpg	Japa Kim 1622211	0.392447	1.27955	Sucesso
Japa_16.jpg	Japa Kim 1622211	0.411494	1.2966	Sucesso
Japa_17.jpg	Japa Kim 1622211	0.421273	1.30867	Sucesso
Japa_19.jpg	Japa Kim 1622211	0.427177	1.2218	Sucesso
Japa_20.jpg	Japa Kim 1622211	0.449316	1.2232	Sucesso
Japa_21.jpg	Japa Kim 1622211	0.554768	1.16041	Sucesso
Japa_22.jpg	Japa Kim 1622211	0.416085	1.79602	Sucesso
Japa_24.jpg	Japa Kim 1622211	0.375422	1.32777	Sucesso
Japa_25.jpg	Japa Kim 1622211	0.392222	1.28808	Sucesso
Japa_26.jpg	Japa Kim 1622211	0.370105	1.29354	Sucesso
Japa_28.jpg	Japa Kim 1622211	0.370145	1.23786	Sucesso
Japa_29.jpg	Japa Kim 1622211	0.35575	1.27384	Sucesso
Japa_30.jpg	Japa Kim 1622211	0.313288	1.25237	Sucesso
Japa_31.jpg	Japa Kim 1622211	0.356227	1.24012	Sucesso
Japa_32.jpg	Japa Kim 1622211	0.385826	1.30597	Sucesso
JapaMan_1.jpg	JapaMan KimK 1234567	0.474248	1.32543	Sucesso
JapaMan_2.jpg	JapaMan KimK 1234567	0.598395	1.27925	Sucesso
JapaMan_3.jpg	JapaMan KimK 1234567	0.410742	1.36353	Sucesso
JapaMan_5.jpg	JapaMan KimK 1234567	0.529443	1.22725	Sucesso
JapaMan_8.jpg	JapaMan KimK 1234567	0.56011	1.33901	Sucesso
JapaMan_9.jpg	JapaMan KimK 1234567	0.460632	1.3306	Sucesso
JapaMan_10.jpg	JapaMan KimK 1234567	0.432079	1.31	Sucesso
JapaMan_11.jpg	JapaMan KimK 1234567	0.433071	1.28443	Sucesso
JapaMan_12.jpg	JapaMan KimK 1234567	0.390608	1.33338	Sucesso
JapaMan_13.jpg	JapaMan KimK 1234567	0.418638	1.30964	Sucesso
JapaMan_15.jpg	JapaMan KimK 1234567	0.556205	1.77075	Sucesso
JapaMan_18.jpg	JapaMan KimK 1234567	0.463739	1.41487	Sucesso
JapaMan_19.jpg	JapaMan KimK 1234567	0.487505	1.89618	Sucesso
JapaMan_21.jpg	JapaMan KimK 1234567	0.439676	1.38762	Sucesso
JapaMan_23.jpg	JapaMan KimK 1234567	0.552039	1.27372	Sucesso
JapaMan_24.jpg	JapaMan KimK 1234567	0.511565	1.29992	Sucesso
JapaMan_25.jpg	JapaMan KimK 1234567	0.497703	1.30774	Sucesso
JapaMan_29.jpg	JapaMan KimK 1234567	0.378683	1.40927	Sucesso
JapaMan_30.jpg	JapaMan KimK 1234567	0.450354	1.59058	Sucesso
JapaMan_31.jpg	JapaMan KimK 1234567	0.318502	1.31566	Sucesso
JapaChild_0.jpg	JapaChild Assaki 1413226	0.582624	1.22103	Sucesso
JapaChild_1.jpg	JapaChild Assaki 1413226	0.568595	1.23682	Sucesso
JapaChild_2.jpg	JapaChild Assaki 1413226	0.519605	1.73678	Sucesso
JapaChild_4.jpg	JapaChild Assaki 1413226	0.244439	1.15062	Sucesso
JapaChild_5.jpg	JapaChild Assaki 1413226	0.601061	1.222	Sucesso
JapaChild_6.jpg	JapaChild Assaki 1413226	0.582624	1.21933	Sucesso
JapaChild_7.jpg	JapaChild Assaki 1413226	0.642181	1.26282	Sucesso
JapaChild_9.jpg	JapaChild Assaki 1413226	0.601061	1.22227	Sucesso
JapaChild_10.jpg	JapaChild Assaki 1413226	0.551592	1.19138	Sucesso
JapaChild_11.jpg	JapaChild Assaki 1413226	0.551592	1.18124	Sucesso
JapaChild_12.jpg	JapaChild Assaki 1413226	0.36469	1.14288	Sucesso
JapaChild_13.jpg	JapaChild Assaki 1413226	0.583093	1.21439	Sucesso
JapaChild_14.jpg	JapaChild Assaki 1413226	0.343405	1.2546	Sucesso
JapaChild_16.jpg	JapaChild Assaki 1413226	0.526704	1.20762	Sucesso
JapaChild_17.jpg	JapaChild Assaki 1413226	0.564543	1.2202	Sucesso
JapaChild_19.jpg	JapaChild Assaki 1413226	0.567856	1.26292	Sucesso
JapaChild_20.jpg	JapaChild Assaki 1413226	0.468304	1.1372	Sucesso
JapaChild_23.jpg	JapaChild Assaki 1413226	0.573665	1.21375	Sucesso
JapaChild_24.jpg	JapaChild Assaki 1413226	0.582624	1.22667	Sucesso
JapaChild_25.jpg	JapaChild Assaki 1413226	0.568559	1.23416	Sucesso
JapaChild_26.jpg	JapaChild Assaki 1413226	0.583093	1.25502	Sucesso
JapaChild_28.jpg	JapaChild Assaki 1413226	0.642181	1.92934	Sucesso
JapaChild_29.jpg	JapaChild Assaki 1413226	0.524968	1.15847	Sucesso
JapaChild_31.jpg	JapaChild Assaki 1413226	0.514311	1.16941	Sucesso
JapaChild_32.jpg	JapaChild Assaki 1413226	0.537096	1.20651	Sucesso
JapaChild_33.jpg	JapaChild Assaki 1413226	0.472778	1.1875	Sucesso
JapaChild_34.jpg	JapaChild Assaki 1413226	0.582624	1.2308	Sucesso

JapaNew_0.jpg	JapaNew Assaki 1223226	0.667111	1.20982	Sucesso
JapaNew_2.jpg	JapaNew Assaki 1223226	0.650356	1.20846	Sucesso
JapaNew_3.jpg	JapaNew Assaki 1223226	0.630966	1.90059	Sucesso
JapaNew_4.jpg	JapaNew Assaki 1223226	0.630966	1.14929	Sucesso
JapaNew_5.jpg	JapaNew Assaki 1223226	0.51853	1.15095	Sucesso
JapaNew_7.jpg	JapaNew Assaki 1223226	0.581337	1.2323	Sucesso
JapaNew_8.jpg	JapaNew Assaki 1223226	0.581337	1.22178	Sucesso
JapaNew_11.jpg	JapaNew Assaki 1223226	0.667111	1.19231	Sucesso
JapaNew_12.jpg	JapaNew Assaki 1223226	0.673094	1.1845	Sucesso
JapaNew_13.jpg	JapaNew Assaki 1223226	0.639006	1.72397	Sucesso
JapaNew_15.jpg	JapaNew Assaki 1223226	0.522185	1.22954	Sucesso
JapaNew_16.jpg	JapaNew Assaki 1223226	0.545831	1.2329	Sucesso
JapaNew_17.jpg	JapaNew Assaki 1223226	0.563366	1.21573	Sucesso
JapaNew_18.jpg	JapaNew Assaki 1223226	0.563366	1.22026	Sucesso
JapaNew_19.jpg	JapaNew Assaki 1223226	0.598778	1.23517	Sucesso
JapaNew_21.jpg	JapaNew Assaki 1223226	0.637358	1.23937	Sucesso
JapaNew_22.jpg	JapaNew Assaki 1223226	0.624964	1.21053	Sucesso
JapaNew_23.jpg	JapaNew Assaki 1223226	0.642381	1.16978	Sucesso
JapaNew_24.jpg	JapaNew Assaki 1223226	0.630207	1.16955	Sucesso
JapaNew_25.jpg	JapaNew Assaki 1223226	0.459535	1.15466	Sucesso
JapaNew_26.jpg	JapaNew Assaki 1223226	0.626095	1.23294	Sucesso
JapaNew_28.jpg	JapaNew Assaki 1223226	0.629369	1.21558	Sucesso
JapaNew_29.jpg	JapaNew Assaki 1223226	0.629369	1.70212	Sucesso
JapaNew_30.jpg	JapaNew Assaki 1223226	0.522098	1.19797	Sucesso
JapaNew_31.jpg	JapaNew Assaki 1223226	0.545831	1.25594	Sucesso
JapaNew_32.jpg	JapaNew Assaki 1223226	0.650356	1.20217	Sucesso
JapaFat_0.jpg	JapaWhite Hourish 2963226	0.602244	1.22163	Falhou
JapaFat_1.jpg	JapaWhite Hourish 2963226	0.602244	1.24146	Falhou
JapaFat_2.jpg	JapaWhite Hourish 2963226	0.602244	1.24848	Falhou
JapaFat_3.jpg	JapaChild Assaki 1413226	0.561051	1.29018	Falhou
JapaFat_6.jpg	JapaWhite Hourish 2963226	0.564662	1.26227	Falhou
JapaFat_7.jpg	JapaWhite Hourish 2963226	0.564662	1.98428	Falhou
JapaFat_10.jpg	JapaChild Assaki 1413226	0.57888	1.24277	Falhou
JapaFat_11.jpg	JapaChild Assaki 1413226	0.511393	1.23464	Falhou
JapaFat_14.jpg	JapaNew Assaki 1223226	0.511393	1.23464	Falhou
JapaFat_15.jpg	JapaNew Assaki 1223226	0.594799	1.23551	Falhou
JapaFat_20.jpg	JapaChild Assaki 1413226	0.594799	1.23121	Falhou
JapaFat_21.jpg	JapaChild Assaki 1413226	0.56793	1.23834	Falhou
JapaFat_22.jpg	JapaChild Assaki 1413226	0.56793	1.23762	Falhou
JapaFat_23.jpg	JapaChild Assaki 1413226	0.570267	1.25737	Falhou
JapaFat_25.jpg	JapaNew Assaki 1223226	0.458574	1.23918	Falhou
JapaFat_27.jpg	JapaNew Assaki 1223226	0.458574	1.21392	Falhou
JapaFat_28.jpg	JapaNew Assaki 1223226	0.553982	1.1803	Falhou
JapaFat_29.jpg	JapaWhite Hourish 2963226	0.615522	1.42293	Falhou
JapaFat_31.jpg	JapaChild Assaki 1413226	0.554089	1.2457	Falhou
JapaFat_32.jpg	JapaWhite Hourish 2963226	0.62542	1.24161	Falhou
JapaFat_34.jpg	JapaWhite Hourish 2963226	0.62542	1.28146	Falhou
JapaUS_1.jpg	Unknown	0.707943	0.970248	Falhou
JapaUS_2.jpg	JapaUS Nashira 4963226	0.656741	1.25312	Sucesso
JapaUS_3.jpg	JapaUS Nashira 4963226	0.656741	1.2519	Sucesso
JapaUS_4.jpg	JapaUS Nashira 4963226	0.655285	1.27851	Sucesso
JapaUS_5.jpg	JapaUS Nashira 4963226	0.655285	1.31464	Sucesso
JapaUS_7.jpg	JapaUS Nashira 4963226	0.625901	1.28241	Sucesso
JapaUS_8.jpg	JapaUS Nashira 4963226	0.644889	1.29286	Sucesso
JapaUS_9.jpg	JapaUS Nashira 4963226	0.598045	1.32806	Sucesso
JapaUS_10.jpg	JapaUS Nashira 4963226	0.692207	1.28739	Sucesso
JapaUS_12.jpg	JapaUS Nashira 4963226	0.635101	1.27695	Su

JapaCN_34.jpg	JapaCN Nawan 7963226	0.659816	1.28804	Sucesso
JapaWhite_0.jpg	JapaWhite Hourish 2963226	0.432055	1.29458	Sucesso
JapaWhite_2.jpg	JapaWhite Hourish 2963226	0.460985	1.2699	Sucesso
JapaWhite_3.jpg	JapaWhite Hourish 2963226	0.400579	1.28046	Sucesso
JapaWhite_4.jpg	JapaWhite Hourish 2963226	0.44452	1.2658	Sucesso
JapaWhite_5.jpg	JapaWhite Hourish 2963226	0.48446	1.32682	Sucesso
JapaWhite_7.jpg	JapaWhite Hourish 2963226	0.465976	1.25794	Sucesso
JapaWhite_8.jpg	JapaWhite Hourish 2963226	0.367916	1.25937	Sucesso
JapaWhite_9.jpg	JapaWhite Hourish 2963226	0.371752	1.26144	Sucesso
JapaWhite_10.jpg	JapaWhite Hourish 2963226	0.371752	1.29482	Sucesso
JapaWhite_12.jpg	JapaWhite Hourish 2963226	0.435674	1.2691	Sucesso
JapaWhite_14.jpg	JapaWhite Hourish 2963226	0.432055	1.28255	Sucesso
JapaWhite_15.jpg	JapaWhite Hourish 2963226	0.435674	1.56385	Sucesso
JapaWhite_18.jpg	JapaWhite Hourish 2963226	0.313192	1.45687	Sucesso
JapaWhite_19.jpg	JapaWhite Hourish 2963226	0.248195	1.23514	Sucesso
JapaWhite_20.jpg	JapaWhite Hourish 2963226	0.485527	1.26338	Sucesso
JapaWhite_22.jpg	JapaWhite Hourish 2963226	0.432055	1.26717	Sucesso
JapaWhite_23.jpg	JapaWhite Hourish 2963226	0.399878	1.27329	Sucesso
JapaWhite_24.jpg	JapaWhite Hourish 2963226	0.359767	1.28245	Sucesso
JapaWhite_25.jpg	JapaWhite Hourish 2963226	0.472319	1.31511	Sucesso
JapaWhite_27.jpg	JapaWhite Hourish 2963226	0.313192	1.46983	Sucesso
JapaWhite_28.jpg	JapaWhite Hourish 2963226	0.359767	1.25701	Sucesso
JapaWhite_29.jpg	JapaWhite Hourish 2963226	0.484005	1.26509	Sucesso
JapaWhite_30.jpg	JapaWhite Hourish 2963226	0.400579	1.26685	Sucesso
JapaWhite_32.jpg	JapaWhite Hourish 2963226	0.248195	1.2515	Sucesso
JapaWhite_33.jpg	JapaWhite Hourish 2963226	0.359767	1.28152	Sucesso
JapaWhite_34.jpg	JapaWhite Hourish 2963226	0.248195	1.2712	Sucesso
JapaOld_0.jpg	JapaOld Akira 1763226	0.529076	1.4307	Sucesso
JapaOld_2.jpg	JapaOld Akira 1763226	0.532657	1.31238	Sucesso
JapaOld_3.jpg	JapaOld Akira 1763226	0.471977	1.31891	Sucesso
JapaOld_4.jpg	JapaOld Akira 1763226	0.493578	1.34388	Sucesso
JapaOld_6.jpg	JapaOld Akira 1763226	0.532657	1.97445	Sucesso
JapaOld_8.jpg	JapaOld Akira 1763226	0.486256	1.40249	Sucesso
JapaOld_9.jpg	JapaOld Akira 1763226	0.51115	1.34489	Sucesso
JapaOld_10.jpg	JapaOld Akira 1763226	0.511844	1.37857	Sucesso
JapaOld_12.jpg	JapaOld Akira 1763226	0.494006	1.3531	Sucesso
JapaOld_13.jpg	JapaOld Akira 1763226	0.578758	1.27735	Sucesso
JapaOld_14.jpg	JapaOld Akira 1763226	0.519614	1.32475	Sucesso
JapaOld_15.jpg	JapaOld Akira 1763226	0.529076	1.40111	Sucesso
JapaOld_17.jpg	JapaOld Akira 1763226	0.492271	1.33242	Sucesso
JapaOld_18.jpg	JapaOld Akira 1763226	0.485922	1.36816	Sucesso
JapaOld_19.jpg	JapaOld Akira 1763226	0.511546	1.3271	Sucesso
JapaOld_21.jpg	JapaOld Akira 1763226	0.561856	1.31583	Sucesso
JapaOld_22.jpg	JapaOld Akira 1763226	0.561856	1.29581	Sucesso
JapaOld_23.jpg	JapaOld Akira 1763226	0.521134	1.36891	Sucesso
JapaOld_25.jpg	JapaOld Akira 1763226	0.590473	1.3231	Sucesso
JapaOld_27.jpg	JapaOld Akira 1763226	0.566533	2.05464	Sucesso
JapaOld_29.jpg	JapaOld Akira 1763226	0.545886	1.36484	Sucesso
JapaOld_30.jpg	JapaOld Akira 1763226	0.529076	1.42977	Sucesso
JapaOld_32.jpg	JapaOld Akira 1763226	0.590473	1.34608	Sucesso
JapaOld_33.jpg	JapaOld Akira 1763226	0.521134	1.34852	Sucesso
JapaOld_34.jpg	JapaOld Akira 1763226	0.519614	1.33364	Sucesso
JapaShort_0.jpg	JapaShort Okuri 1232266	0.443096	1.3031	Sucesso
JapaShort_2.jpg	JapaShort Okuri 1232266	0.421989	1.95653	Sucesso
JapaShort_4.jpg	JapaShort Okuri 1232266	0.51154	1.33918	Sucesso
JapaShort_5.jpg	JapaShort Okuri 1232266	0.617195	1.25277	Sucesso
JapaShort_6.jpg	JapaShort Okuri 1232266	0.424969	1.28549	Sucesso
JapaShort_7.jpg	JapaShort Okuri 1232266	0.424969	1.30477	Sucesso
JapaShort_9.jpg	JapaShort Okuri 1232266	0.352138	1.25883	Sucesso
JapaShort_10.jpg	JapaShort Okuri 1232266	0.352501	1.27933	Sucesso
JapaShort_11.jpg	JapaShort Okuri 1232266	0.347619	1.28278	Sucesso
JapaShort_12.jpg	JapaShort Okuri 1232266	0.412218	1.30596	Sucesso
JapaShort_15.jpg	JapaShort Okuri 1232266	0.369614	1.30553	Sucesso
JapaShort_16.jpg	JapaShort Okuri 1232266	0.379628	1.28066	Sucesso

JapaShort_17.jpg	JapaShort Okuri 1232266	0.337615	1.88754	Sucesso
JapaShort_19.jpg	JapaShort Okuri 1232266	0.353686	1.41389	Sucesso
JapaShort_20.jpg	JapaShort Okuri 1232266	0.384501	1.27012	Sucesso
JapaShort_21.jpg	JapaShort Okuri 1232266	0.544987	1.27335	Sucesso
JapaShort_23.jpg	JapaShort Okuri 1232266	0.373197	1.29763	Sucesso
JapaShort_24.jpg	JapaShort Okuri 1232266	0.323678	1.29265	Sucesso
JapaShort_25.jpg	JapaShort Okuri 1232266	0.387505	1.2864	Sucesso
JapaShort_26.jpg	JapaShort Okuri 1232266	0.501975	1.27185	Sucesso
JapaShort_28.jpg	JapaShort Okuri 1232266	0.397541	1.28206	Sucesso
JapaShort_29.jpg	JapaShort Okuri 1232266	0.352795	1.27875	Sucesso
JapaShort_30.jpg	JapaShort Okuri 1232266	0.369614	1.31208	Sucesso
JapaShort_31.jpg	JapaShort Okuri 1232266	0.443098	1.32219	Sucesso
JapaHair_1.jpg	JapaWhite Hourish 2963226	0.626232	1.27017	Falhou
JapaHair_3.jpg	JapaUS Nashira 4963226	0.583009	1.27828	Falhou
JapaHair_4.jpg	JapaUS Nashira 4963226	0.574655	1.26663	Falhou
JapaHair_5.jpg	JapaShort Okuri 1232266	0.508948	1.25561	Falhou
JapaHair_6.jpg	JapaWhite Hourish 2963226	0.395863	1.35481	Falhou
JapaHair_8.jpg	JapaWhite Hourish 2963226	0.423955	1.24523	Falhou
JapaHair_9.jpg	JapaWhite Hourish 2963226	0.433921	1.2322	Falhou
JapaHair_10.jpg	JapaWhite Hourish 2963226	0.537979	1.24185	Falhou
JapaHair_11.jpg	JapaUS Nashira 4963226	0.654008	1.28939	Falhou
JapaHair_12.jpg	JapaUS Nashira 4963226	0.632239	1.27354	Falhou
JapaHair_14.jpg	JapaUS Nashira 4963226	0.602046	1.24307	Falhou
JapaHair_15.jpg	JapaUS Nashira 4963226	0.609358	1.23308	Falhou
JapaHair_16.jpg	JapaWhite Hourish 2963226	0.550213	1.25183	Falhou
JapaHair_17.jpg	JapaUS Nashira 4963226	0.641139	1.27373	Falhou
JapaHair_19.jpg	JapaUS Nashira 4963226	0.641426	1.25323	Falhou
JapaHair_20.jpg	JapaUS Nashira 4963226	0.609417	1.22834	Falhou
JapaHair_23.jpg	JapaShort Okuri 1232266	0.355387	1.27041	Falhou
JapaHair_24.jpg	JapaWhite Hourish 2963226	0.462563	1.22464	Falhou
JapaHair_25.jpg	JapaWhite Hourish 2963226	0.488596	1.22814	Falhou
JapaHair_26.jpg	JapaFun Nagawa 3963226	0.521169	1.29372	Falhou
JapaHair_28.jpg	JapaUS Nashira 4963226	0.657668	1.27897	Falhou
JapaLong_0.jpg	Unknown	0.735998	0.990609	Falhou
JapaLong_1.jpg	Unknown	0.754843	0.981802	Falhou
JapaLong_2.jpg	Unknown	0.72343	1.00726	Falhou
JapaLong_3.jpg	JapaLong Oishi 1232257	0.586901	1.67098	Sucesso
JapaLong_6.jpg	Unknown	0.771221	1.1521	Falhou
JapaLong_7.jpg	Unknown	0.701866	0.970426	Falhou
JapaLong_8.jpg	Unknown	0.782755	1.32228	Falhou
JapaLong_9.jpg	Unknown	0.828582	1.21352	Falhou
JapaLong_10.jpg	Unknown	0.758865	0.982342	Falhou
JapaLong_11.jpg	JapaLong Oishi 1232257	0.560491	1.29837	Sucesso
JapaLong_12.jpg	JapaLong Oishi 1232257	0.68782	1.23449	Sucesso
JapaLong_14.jpg	JapaLong Oishi 1232257	0.564448	1.25918	Sucesso
JapaLong_15.jpg	JapaLong Oishi 1232257	0.569404	1.25614	Sucesso
JapaLong_16.jpg	JapaLong Oishi 1232257	0.528762	1.2522	Sucesso
JapaLong_17.jpg	JapaLong Oishi 1232257	0.573949	1.28266	Sucesso
JapaLong_21.jpg	Unknown	0.763377	1.06874	Falhou
JapaLong_22.jpg	Unknown	0.834504	0.956507	Falhou
JapaLong_23.jpg	Unknown	0.75984	1.08684	Falhou
JapaLong_24.jpg	Unknown	0.759223	0.986327	Falhou
JapaLong_25.jpg	Unknown	0.759223	0.906309	Falhou
JapaLong_26.jpg	JapaLong Oishi 1232257	0.640719	1.28091	Sucesso
JapaLong_28.jpg	Unknown	0.771221	1.49662	Falhou
JapaLong_29.jpg	JapaLong Oishi 1232257	0.569404	1.45477	Sucesso
JapaLong_30.jpg	JapaLong Oishi 1232257	0.569404	1.28178	Sucesso
JapaLong_31.jpg	Unknown	0.763377	0.99287	Falhou
JapaLong_32.jpg	JapaLong Oishi 1232257	0.68782	1.20818	Sucesso

Média 0.54 1.28
Desvio Padrão 0.12 0.17
Total Amostras 418

Falhou 20.6%
Sucesso 79.4%