



Pedro Leandro Moreira Detecção e localização de faces em imagens
Graça



Pedro Leandro Moreira Detecção e localização de faces em imagens Graça

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Prof. Dr. Ana Maria Perfeito Tomé, Professora associada do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e sob a co-orientação científica do Prof. Dr. Luís Filipe de Seabra Lopes, Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri

presidente

Doutor Armando José Formoso de Pinho
Professor associado da Universidade de Aveiro

vogais

Doutora Ana Maria Perfeito Tomé
Professora associada da Universidade de Aveiro

Doutor Luís Filipe de Seabra Lopes
Professor auxiliar da Universidade de Aveiro

Doutor Luís Miguel Almeida da Silva
Professor auxiliar convidado da Universidade do Minho

agradecimentos

À Professora Ana Maria Perfeito Tomé por tudo que me ensinou tanto a nível de docência como a nível pessoal. Por me ter ajudado a perceber a importância da tese e por ter acreditado em mim.

Ao Professor Luís Seabra Lopes por todo o apoio prestado na realização da tese.

A todos os meus colegas e amigos que me ajudaram a evoluir ao longo de todo o percurso universitário tanto a nível profissional como a nível pessoal.

À Nati por todo o apoio que me deu e por me fazer perceber que terminar a tese era mesmo importante.

Para terminar, aos meus pais e irmã, por tudo que representam para mim.

palavras-chave

Detecção e localização de faces, processamento, detecção de zonas de pele, método do protótipo, correlação cruzada, espaço de cores

resumo

Desde a existência do Homem que este utiliza a visão para detectar e localizar faces humanas para identificar e interagir com a face identificada de forma apropriada. Em visão computacional, o simples processo efectuado pelo Homem de detectar e localizar faces em imagens é bastante complexo. Nos últimos anos houve um grande aumento de pesquisa sobre a detecção e localização de faces em imagens, devido ao facto de ser o primeiro passo em qualquer sistema de reconhecimento facial. As inúmeras dificuldades que as características da face apresentam (tamanho, posição, orientação, entre outras) são um desafio para a implementação do algoritmo. Existe um grande número de técnicas para resolver esses problemas.

Neste trabalho foi implementado um algoritmo para detecção e localização de faces em imagens que num primeiro passo consiste na detecção de pele, através de 3 métodos com 3 espaços de cores distintos. Posteriormente é aplicado o método do template a algumas regras geométricas baseadas no conhecimento para detectar se as diferentes zonas de pele detectadas são face ou não-face. Por ultimo é efectuada a localização da face na imagem.

Índice

1 - INTRODUÇÃO	4
1.1 – OBJECTIVOS.....	5
1.2 – ORGANIZAÇÃO.....	6
1.3 – DIAGRAMA DE FLUXO GERAL DO SISTEMA IMPLEMENTADO	6
2 - DETECÇÃO E LOCALIZAÇÃO DE FACES	8
2.1 – SISTEMAS BASEADOS EM BLOCOS	8
2.2 – SISTEMAS BASEADOS NA EXTRACÇÃO CARACTERÍSTICAS FACIAIS.....	9
2.2.1 – <i>Combinação de características faciais</i>	10
2.2.2 – <i>Pele</i>	10
2.2.3 – <i>Textura</i>	11
2.2.4 – <i>Protótipos</i>	12
3 - SISTEMAS BASEADOS EM DETECÇÃO DE PELE	13
3.1 - CONCEITOS BÁSICOS	14
3.1.1 - <i>ImagemDigital</i>	14
3.2 – DIAGRAMA DE BLOCOS DO SISTEMA IMPLEMENTADO	16
3.3 - ESPAÇO DE CORES	18
3.3.1 – <i>RGB</i>	18
3.3.2 – <i>YCrCb</i>	19
3.3.3 – <i>HSV</i>	20
3.4 – DETECTORES DE PELE	22
3.4.1 – <i>Classificadores baseados em regras</i>	22
3.4.1.1 – <i>Modelo RGB</i>	23
3.4.1.2 – <i>Modelo HSV</i>	23
3.4.2 – <i>Classificador Gaussiano</i>	24
3.5 – OPERADORES MORFOLÓGICOS EM IMAGENS BINÁRIAS	26
3.5.1 – <i>Erosão e dilatação</i>	26
3.6 – MÉTODO DO PROTÓTIPO	28
3.6.1 – <i>Protótipo</i>	28
3.6.2 – <i>Correlação cruzada</i>	29
4 – RESULTADOS	34
4.1 – BASES DE DADOS	34
4.2 – INTERFACE.....	34
4.3 – DETECÇÃO DE FACES	35
4.3.1 – <i>Classificador baseado em regras (RGB)</i>	36
4.3.2 – <i>Classificador baseado em regras (HSV)</i>	37
4.3.3 – <i>Classificador Gaussiano</i>	39
4.4 – COMPARAÇÃO DOS DETECTORES DE PELE	40
4.5 – IMAGENS DE FACE COM FUNDO SIMPLES	40
4.6 – IMAGENS DE FACE COM FUNDO COMPLEXO.....	44
4.7 – IMAGENS DE GRUPO	45
5 - CONCLUSÕES	47
6 – TRABALHO FUTURO	49
7 - BIBLIOGRAFIA	50
8. APÊNDICE	54

Lista de Figuras

FIGURA 1.1: DIAGRAMA DE FLUXO GERAL DO ALGORITMO	6
FIGURA 3.1: REPRESENTAÇÃO MATRICIAL DE UMA IMAGEM	15
FIGURA 3.2: DIAGRAMA DE BLOCOS DO ALGORITMO	17
FIGURA 3.3: CUBO RGB	19
FIGURA 3.4: ESPAÇO DE CORES HSV	21
FIGURA 3.5: MODELO DE DISTRIBUIÇÃO JOINT GAUSSIANA PARA O CONJUNTO DE TREINO	25
FIGURA 3.6: APLICAÇÃO DA EROÇÃO A UM CONJUNTO DE PIXELS	27
FIGURA 3.7: APLICAÇÃO DA DILATAÇÃO A UM CONJUNTO DE PIXELS	28
FIGURA 3.8: PROTÓTIPO IMPLEMENTADO	29
FIGURA 3.9: ÂNGULO DA REGIÃO A CLASSIFICAR	30
FIGURA 3.10: IMAGEM ORIGINAL E IMAGEM ORIGINAL INVERTIDA	31
FIGURA 3.11: ZONAS DE PELE DETECTADAS	32
FIGURA 3.12: ZONAS DE PELE DETECTADAS DELIMITADAS POR 1 RECTÂNGULO	32
FIGURA 4.1: MENU DO PROGRAMA	35
FIGURA 4.2: IMAGENS ORIGINAIS	36
FIGURA 4.3: IMAGENS BINÁRIAS COM AS ZONAS DE PELE DETECTADAS	36
FIGURA 4.4: IMAGENS BINÁRIAS DEPOIS DO MÉTODO DA EROÇÃO E DA DILATAÇÃO	36
FIGURA 4.5: AS DUAS REGIÕES DE PELE SEPARADAS	37
FIGURA 4.6: IMAGENS FINAIS COM A DETECÇÃO	37
FIGURA 4.7: IMAGENS BINÁRIAS COM AS ZONAS DE PELE DETECTADAS	37
FIGURA 4.8: IMAGENS BINÁRIA DEPOIS DO MÉTODO DA EROÇÃO E DA DILATAÇÃO	38
FIGURA 4.9: IMAGENS FINAIS COM A DETECÇÃO	38
FIGURA 4.10: IMAGENS BINÁRIAS COM AS ZONAS DE PELE DETECTADAS	39
FIGURA 4.11: IMAGENS BINÁRIA DEPOIS DO MÉTODO DA EROÇÃO E DA DILATAÇÃO	39
FIGURA 4.12: IMAGENS FINAIS COM A DETECÇÃO	39
FIGURA 4.13: FACE MASCULINA, 19 ANOS, ETNIA BRANCA	41
FIGURA 4.14: FACE MASCULINA, 24 ANOS, ETNIA ÍNDIA	41
FIGURA 4.15: FACE FEMININA, 25 ANOS, ETNIA ÍNDIA	41
FIGURA 4.16: FACE FEMININA, 37 ANOS, ETNIA NEGRA	41
FIGURA 4.17: FACE MASCULINA, 50 ANOS, ETNIA NEGRA	41
FIGURA 4.18: FACE FEMININA, 70 ANOS, ETNIA BRANCA	42
FIGURA 4.19: IMAGEM ORIGINAL E IMAGEM FINAL COM A DETECÇÃO	44
FIGURA 4.20: IMAGEM ORIGINAL E IMAGEM FINAL COM A DETECÇÃO	44
FIGURA 4.21: IMAGEM ORIGINAL E IMAGEM FINAL COM A DETECÇÃO	44
FIGURA 4.22: IMAGEM ORIGINAL E IMAGEM FINAL COM A DETECÇÃO	45
FIGURA 4.23: IMAGEM FINAL COM A DETECÇÃO	45
FIGURA 4.24: IMAGEM FINAL COM A DETECÇÃO	46
FIGURA 4.25: IMAGEM FINAL COM A DETECÇÃO	46

Lista de Tabelas

TABELA 4.1: RESULTADOS DOS DIFERENTES DETECTORES DE PELE	40
TABELA 4.2: RESULTADOS DOS DIFERENTES DETECTORES DE PELE (%).....	40
TABELA 4.3: RESULTADOS PARA IMAGENS DE DIFERENTES SEXOS E IDADES	42
TABELA 4.4: RESULTADOS PARA IMAGENS DE DIFERENTES SEXOS E IDADES (%)	42
TABELA 4.5: RESULTADOS PARA IMAGENS DE DIFERENTES SEXOS	43
TABELA 4.6: RESULTADOS PARA IMAGENS DE DIFERENTES SEXOS (%).....	43
TABELA 4.7: RESULTADOS PARA IMAGENS DE DIFERENTES IDADES	43
TABELA 4.8: RESULTADOS PARA IMAGENS DE DIFERENTES IDADES (%)	43

CAPITULO 1

Introdução

O reconhecimento de faces humanas é uma das tecnologias mais estudadas em biometria devido ao facto de não ser necessária qualquer colaboração da pessoa a identificar, ser intuitiva e ser uma técnica não-invasiva. Tem um grande número de aplicações, tais como, sistemas de segurança, identificação e autenticação, interacção de agentes artificiais com utilizadores humanos e visão robótica. Previamente ao reconhecimento é necessário detectar e localizar a face na imagem.

Computacionalmente a detecção e localização de faces em imagens pode ser definido como o processo que tem uma imagem como entrada e produz na saída a mesma imagem com as faces assinaladas. Assim, os resultados do processo de detecção são as coordenadas de áreas rectangulares onde a face pode estar localizada na imagem. Pode também ser considerado como um pré-processamento básico para qualquer sistema de processamento de imagens ou vídeos que interaja com a face humana. Desde o final da década de 80 que o reconhecimento de faces tem vindo a ser estudado, [5], [6] são dois artigos de revisão muito completos.

O grande desafio na construção de um sistema de detecção e localização de faces está nas inúmeras dificuldades em detectar a face, quando a mesma apresenta diferentes posições, expressões faciais, condições de luminosidade, componentes estruturais, oclusões, expressões faciais, condições de imagem e orientações de imagem.

- As imagens de uma face variam consoante a posição relativa camera-face (frontal, quarenta e cinco graus, perfil, entre outras), o que

pode fazer com que os olhos e o nariz fiquem parcialmente ou totalmente ocultos.

- Uma face pode conter componentes estruturais (bigode, barba, óculos e outros adornos) que podem transmitir à imagem uma variância de forma e cor.

- A expressão facial afecta a aparência da face na imagem, a expressão facial pode alterar os contornos da face.

- Uma face pode estar parcialmente ocultada numa imagem por outro objecto. Num grupo de pessoas, uma face pode ocultar parcialmente outra face.

- Imagens de face variam directamente com as diferentes rotações do eixo óptico da camera.

- Quando a imagem é formada, factores como a luminosidade e as características da camera afectam a aparência da face. Uma imagem que apresenta muita iluminação apresenta diferenças nas cores perante a mesma imagem sem a iluminação.

1.1 – Objectivos

Os objectivos principais para este trabalho são o estudo e revisão da bibliografia disponível, identificação dos requisitos do sistema, o desenvolvimento de um classificador em duas classes: face e não-face e demonstrar que o algoritmo funciona para faces de todas as idades, sexo, etnias e com fundos complexos, para finalizar testar o algoritmo em bases de dados públicas.

1.2 – Organização

Esta dissertação divide-se em 5 capítulos. O capítulo 1 faz uma introdução geral ao tema da detecção e localização de faces, referindo a sua utilidade, definindo-o, analisando os problemas na sua construção e explica claramente em que consiste o trabalho.

O capítulo 2 faz uma abordagem aos sistemas de detecção de faces e faz referência a alguns trabalhos importantes já desenvolvidos.

O capítulo 3 faz uma abordagem aos sistemas baseados em detecção de pele e faz referência a alguns trabalhos importantes já desenvolvidos. Posteriormente aborda a imagem teoricamente e apresenta uma explicação detalhada de todo o trabalho desenvolvido.

O capítulo 4 apresenta todos os resultados, assim como, algumas considerações que levaram a tomar algumas opções no desenvolvimento do algoritmo.

O capítulo 5 apresenta as conclusões do trabalho desenvolvido, apontando as potencialidades do algoritmo desenvolvido e as suas falhas.

1.3 – Diagrama de fluxo geral do sistema implementado



Figura 1.1 – Diagrama de fluxo geral do algoritmo

Na figura 1.1 é possível observar um diagrama de fluxo muito geral do algoritmo que se desenvolveu. O algoritmo divide-se em três grandes blocos, o primeiro bloco é a Detecção de Pele, tem como parâmetro de entrada uma imagem colorida e como parâmetro de saída uma imagem binária, onde o branco representa um pixel com pele e o preto representa um pixel não-Pele. O segundo bloco é a Detecção da face e tem como parâmetro de entrada uma imagem binária com todas as zonas de pele do corpo e como parâmetro de saída uma imagem binária com as faces detectadas. O terceiro bloco tem como parâmetro de entrada uma imagem binária com as faces detectadas e tem como parâmetro de saída a imagem colorida inicial com um rectângulo a identificar a face.

Capítulo 2

Detecção e localização de faces

Os sistemas baseados em detecção e localização de faces em imagens podem ser divididos em sistemas baseados em blocos de imagem e sistemas baseados em extracção de características faciais. Seguidamente é feita a definição de ambos os sistemas e são referidos alguns trabalhos já desenvolvidos.

2.1 – Sistemas baseados em blocos de imagem

Sistemas baseados em blocos de imagem tratam a detecção de faces como um problema de classificação que está relacionado com o treino e a aprendizagem do sistema. Estes sistemas necessitam que seja efectuada uma aprendizagem sobre as características da face a partir de um grupo de blocos que servem de treino. Cada bloco possui uma imagem unicamente da face (do queixo até à testa).

Em seguida são referidos os principais trabalhos que se realizaram nos últimos anos em sistemas baseados em blocos.

Em [38] é proposto um algoritmo baseado em aprendizagem através do agrupamento de blocos (19X19 pixels) que modelam a distribuição de exemplos de faces e de não-face. Através da distância definidas em comparação com o modelo de distribuição, é treinado um classificador perceptrão multicamada para determinar a presença ou ausência de faces em regiões da imagem

Rowley [1], [2] propôs uma técnica baseada numa rede neural multicamada. As classes face e não face são aprendidas no conjunto de treino. Neste trabalho foi focado o problema de a face poder aparecer na imagem com diferentes escalas. A resolução do problema é feita através da análise de imagens multi-resolução e definindo um mínimo para o tamanho da face (19x19 pixels), de forma a reduzir o espaço de procura. Em relação à posição relativa camera-face, desenvolveu um algoritmo para detectar faces frontais [1] e seguidamente evoluiu o algoritmo de forma a detectar também faces com diferentes níveis de rotação [2].

Samal and Iyengar [31] implementaram um sistema de detecção, facial baseado na técnica do PCA (análise dos componentes principais), [12] baseado na projecção das faces. Em vez de faces próprias eles obtiveram a projecção das faces próprias e efectuaram uma combinação com técnicas de processamento e a transformada de Hough.

2.2 – Sistemas baseados na extracção características faciais

Alguns sistemas efectuam a extracção de componentes faciais com o objectivo de determinar a existência ou não da face.

Em seguida são referidos os principais trabalhos que se realizaram nos últimos anos em sistemas baseados na extracção de características faciais. São abordadas as principais características faciais. Para uma consulta muito detalhada de todas as características faciais usadas, consultar os artigos de revisão [5] e [6]

2.2.1 – Combinação de características faciais

Yang e Huang [28] em 1994 implementaram um sistema de extracção de características faciais que se dividia em 3 fases. Na primeira fase eram localizadas todas as possíveis imagens através de regras que consistiam em descrições generalizadas sobre o aspecto da face. Na segunda fase eram usados histogramas e métodos de equiparação para detectar as bordas. Na última fase eram detectadas as características faciais (olhos e boca), baseando-se em um conjunto de regras que correspondem às mesmas.

2.2.2 – Pele

Os sistemas baseados na característica facial pele necessitam de outro método para, posteriormente à detecção de pele, efectuar a eliminação das diferentes zonas de pele (braço, perna, mão e pescoço) que não interessam, efectuando assim a localização da face. Em seguida são referidos trabalhos importantes no âmbito da detecção de pele.

As principais características faciais invariantes que permitem uma boa detecção da face são a cor de pele e a textura da face segundo Yang [5].

O sistema [35] mais simples para detectar as zonas de pele consiste em definir um pixel de pele com os valores das componentes de crominância Cr e Cb obtidos a partir das amostras de pele.

Em [14] é feita a comparação de cinco espaços de cores e dois métodos não-paramétricos de decisão para detecção de pele. Foi concluído que para uma aproximação bayseana, a escolha do espaço de cores não

influencia no resultado. Em relação ao método LUT (olhar sobre a tabela) o espaço de cores HSV apresenta melhores resultados.

Em [15] é feita uma comparação entre nove espaços de crominância e de dois métodos paramétricos (Gaussiana simples e mistura de gaussianas) de decisão para detectar pele concluindo que a mistura de Gaussianas apresenta melhores resultados.

Um estudo efectuado [6] analisa a importância da etnia na detecção de zonas de pele concluindo que a pele de pessoas com diferente etnia possuem diferente luminosidade e crominância igual.

Em [18] é efectuado um estudo sobre a interferência dos vários espaços de cores na detecção de zonas de pele, onde se conclui que o espaço de cores não interfere na detecção das zonas de pele.

2.2.3 - Textura

A textura única da face humana permite efectuar uma eficaz separação da face relativamente ao fundo da imagem.

Augusteijn e Skufca [6] desenvolveram um sistema que efectua a detecção de faces a partir da textura da face. Usaram o modelo SGDL [39] para detectar a face. O modelo consiste em transformar a textura para a forma computacional usando características faciais de segunda ordem em blocos de imagem 16X16. Inicialmente efectuaram a extracção de 3 características (pele, cabelo e outro). Para efectuar uma classificação supervisionada das diferentes texturas usaram uma rede de correlação neural [40]. Para agrupar as diferentes classes de textura usaram um mapa de Kohonen [41].

2.2.4 – Protótipos

As técnicas baseadas no emparelhamento de modelos protótipos [30] são técnicas que se baseiam na representação da face através de um modelo padrão. Efectuando a correlação entre o protótipo e a face presente numa imagem é possível obter a detecção.

Sakai [36] utilizou uma técnica baseada em modelos protótipos para detectar faces. Utilizou vários sub-protótipos (olho, nariz, boca e bordas da face). Cada sub-protótipos foi implementado com linhas segmentadas. Eram retiradas as bordas da imagem de entrada e posteriormente eram comparadas com o sub-protótipos (bordas da face). Através da correlação entre as duas imagens e as bordas das imagens eram detectadas as possíveis localizações das faces. E finalmente era feita a comparação com os restantes sub-protótipos para obter apenas a face e eliminar as regiões mal detectadas na primeira fase. Esta técnica foi desenvolvida em 1969, mas ainda hoje tem uma grande aceitação por parte dos investigadores segundo Yang [5].

Capítulo 3

Sistemas baseados em detecção de pele

A detecção de pele em imagens coloridas é uma técnica muito popular e eficaz na detecção e localização de faces em imagens. É uma técnica que tem vindo a ser muito pesquisada nas últimas décadas.

O método de detecção das zonas de pele [4] [8] é o método explorado neste trabalho. Consiste na resolução de três problemas. O primeiro é a determinação do espaço de cores a usar. O segundo é descobrir como deve ser modelada exactamente a distribuição colorida da pele. Por ultimo, o processamento da segmentação dos resultados obtidos anteriormente para determinar se a região de pele obtida é uma face ou uma não face (braço, mão, perna, entre outros).

Assim é possível definir computacionalmente a detecção de zonas de pele em imagens coloridas [37], [3] através da análise da imagem pixel a pixel ou região a região.

Na abordagem pixel a pixel, o classificador percorre a imagem pixel a pixel e independentemente dos vizinhos classifica o pixel como pele ou não-pele.

Em relação ao método região a região, é necessário ter em consideração o arranjo espacial dos pixels de pele durante a detecção para melhorar os métodos de desempenho.

Nesta dissertação irá ser abordada apenas a detecção de zonas de pele pixel a pixel.

Um artigo muito importante onde se pode entender perfeitamente o processo da detecção e localização de zonas de pele pixel a pixel em imagens coloridas é o artigo de revisão de Vezhnevets et al [13].

3.1 - Conceitos básicos

Neste ponto são apresentados os conceitos introdutórios à imagem digital, assim como, as possíveis representações. É apresentada uma definição de imagem digital, é representada matricialmente a imagem digital e são referidas as suas possíveis representações.

3.1.1 - Imagem Digital

Uma imagem é a representação estática de um objecto obtido com um sistema de produção de imagens.

Uma imagem digital pode ser descrita matematicamente pela função bidimensional de intensidade luminosa f (linha, coluna), onde cada valor de f é proporcional à luminosidade da imagem nesse ponto para cada coordenada espacial (linha, coluna) [23].

Uma imagem digital pode ser representada por uma matriz (figura 3.1) onde os índices das linhas e das colunas identificam um ponto na imagem e o correspondente valor representa a luminosidade. Cada um desses valores ($f(\text{linha}, \text{coluna})$) é denominado por pixel ou elemento da imagem.

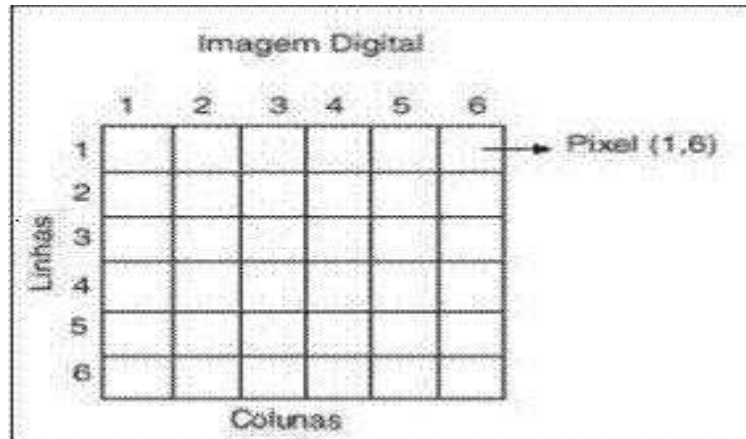


Figura 3.1 – Representação matricial de uma imagem

As imagens digitais podem ser a preto e branco (binária), níveis de cinzento, cor indexada e de cor (RGB por exemplo).

Em imagens a preto e branco, os pixels da imagem são representados pelo valor da cor preta e branca.

Em imagens com níveis de cinzento, a representação dos pixels da imagem são as cores preto e branco e uma gama de cinza. As imagens podem conter 4 bits por pixel (16 tons de cinza) e imagens de 8 bits por pixel possuem um valor de brilho.

Imagens de cor indexada são representadas com 4 bits por pixel (16 cores) ou 8 bits por pixel (256 cores) e as suas cores encontram-se definidas numa tabela de valores de cores.

Imagens de cores são representadas com 24 bits por pixel. No caso do modelo RGB as cores são divididas em 8 bits para o vermelho, 8 para o verde e 8 para o azul.

Nesta dissertação apenas serão analisadas as imagens a cores, na próxima secção serão analisados mais detalhadamente os principais espaços de cores usados na detecção e localização de faces em imagens.

3.2 – Diagrama de blocos do sistema implementado

Este ponto consiste na apresentação do diagrama de blocos específico do sistema implementado.

Na figura 3.2 é possível observar um diagrama de blocos onde é feita uma demonstração do algoritmo a realizar. Ao longo da dissertação cada bloco irá ser explicado detalhadamente. O bloco 1 é explicado detalhadamente no ponto 3.3, o bloco 2 no ponto 3.4, o bloco 3 no ponto 3.5 e os blocos 4,5,6 e 7 no ponto 3.6.

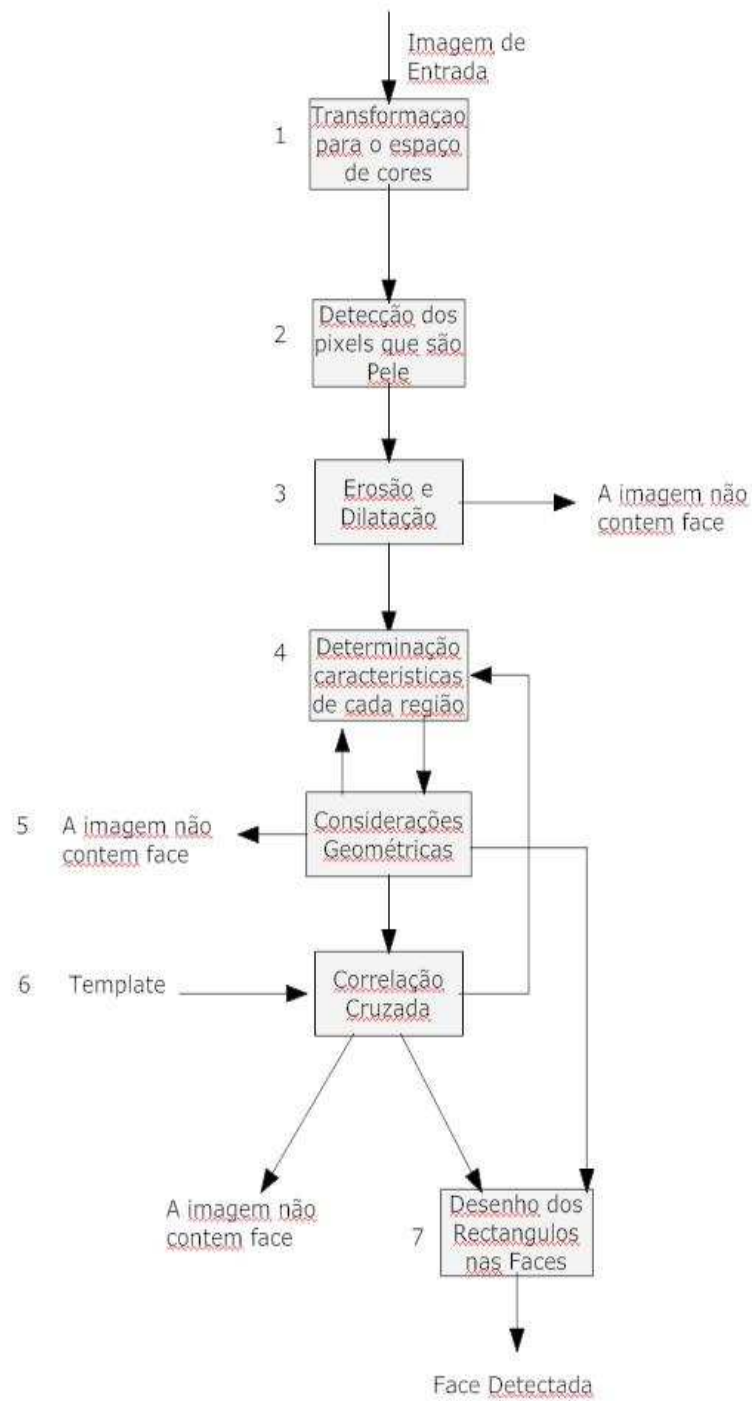


Figura 3.2 – Diagrama de blocos do algoritmo

3.3 - Espaço de cores

O objectivo de um espaço de cores é facilitar a representação das cores que compõem uma imagem através de um padrão. Basicamente um espaço de cores corresponde a um sub-espaço do sistema cartesiano tridimensional onde cada cor é representada por um ponto.

3.3.1 – RGB

O modelo de espaço de cores RGB é um espaço de cores com uma grande aceitação entre os modelos de cor. Foi estudada [27] a separabilidade do espaço de cores. Modelo inventado por Thomas Young (1773 – 1829), este modelo tem como principio a combinação do vermelho, verde e azul. Combinando as cores de formas diferentes pode obter-se qualquer cor. É um dos espaços de cor mais usados para processamento e armazenamento de dados de imagens digitais. A alta correlação entre canais, uma significativa não-uniformidade e uma mistura de cromaticidade e luminosidade nos dados faz com que o RGB não seja uma escolha muito favorável para a análise da cor e para algoritmos de reconhecimento baseados em cor [13].

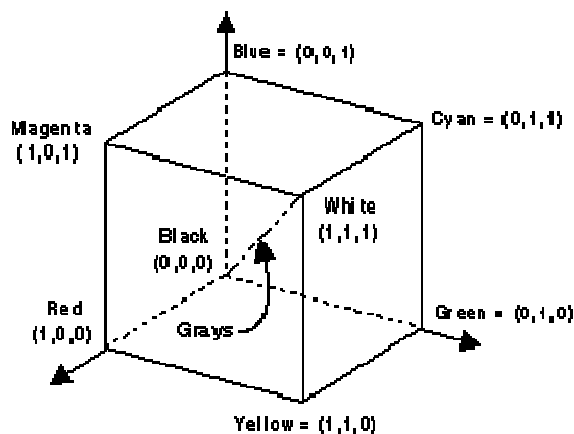


Figura 3.3 – Cubo RGB

Na figura 3.3 é possível observar o cubo RGB, que é uma representação da distribuição de cores RGB. Foi construído para que cada ponto tenha o valor RGB correspondente. Por exemplo o ponto $(0, 0, 1)$ corresponde ao valor RGB $(0, 0, 255)$, que representa a cor azul.

O espaço RGB permite a obtenção do espaço de cores RGB normalizado. Tem como base que a soma das três componentes normalizada é igual a um ($R+B+G=1$). Tendo em conta que a soma das três componentes normalizadas é um, a terceira componente pode ser omitida e conseqüentemente reduzir o espaço dimensional [24]. É um método com bastante aceitação devido à sua simplicidade aliada ao facto da obtenção de bons resultados.

3.3.2 – YCrCb

O espaço de cores YCrCb é uma codificação do espaço de cores não linear RGB. O Y é a componente de luminosidade, o Cr e o Cb são as componentes de cromaticidade. Este espaço de cores efectua a separação

entre a luminosidade e as componentes de cor e é muito usado em compressão de imagens. A passagem do espaço de cores de RGB para YCrCb permite eliminar o efeito da luminosidade. Os valores de Cr e Cb são obtidos através da subtração da luminosidade a cada uma das componentes.

$$Y = 0.299R + 0,587G + 0.114B \quad (3.1)$$

$$Cr = R - Y \quad (3.2)$$

$$Cb = B - Y \quad (3.3)$$

3.3.3 – HSV

O espaço de cores HSV possui três componentes que são o matiz (H), a saturação (S) e a luminosidade (V). O matiz está relacionada com o comprimento espectral da onda que produz a percepção da cor, a saturação está relacionada com a pureza da cor (uma cor é pura quando possui poucas combinações de diferentes cores) e por ultimo a luminosidade está associada ao brilho da cor.

[25] É aplicado um método que retira os pontos indesejáveis do espaço de cores, incluindo as descontinuidades de saturação e a computação da luminosidade (V), que afectavam negativamente as propriedades das cores:

$$H = \arccos \frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{((R - G)^2 + (R - B)(G - B))}} \quad (3.4)$$

$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B} \quad (3.5)$$

$$V = \frac{1}{3}(R + G + B) \quad (3.6)$$

A representação gráfica do espaço de cores HSV é um cone tridimensional de 6 lados derivado do cubo RGB. Os vários matizes encontram-se na parte superior do cone. O matiz varia entre 0 e 360 graus. A saturação varia entre 0 e 1 sendo representada pela razão entre a pureza de um determinado matiz e a sua pureza máxima (quando $S = 1$). A luminosidade varia de 0 (pico do cone), que representa a cor preta, a 1 (base), onde as intensidades das cores são máximas. Esses valores foram representados [26] como pode ser observado na figura 3.4.

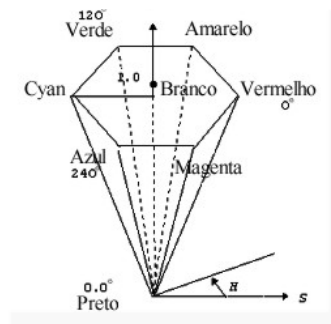


Figura 3.4 – Espaço de cores HSV

3.4 – Detectores de pele

O objectivo dos detectores de pele propostos é receber como parâmetro de entrada a imagem original e classificar cada pixel em pele ou não-pele e conseqüentemente obter uma imagem binária onde o branco representa a pele e o preto não-pele.

Neste ponto são descritos detalhadamente 3 métodos para detectar os pixels de pele numa imagem.

3.4.1 – Classificadores baseados em regras

Nestes métodos as regiões de pele são obtidas através de um conjunto de regras que permite obter as margens das regiões de pele. Há uma grande aceitação devido ao facto de ser um método de muito simples execução e conseqüentemente levar à construção de um algoritmo rápido aliado a bons resultados.

Os classificadores baseados em regras utilizam regras pré-definidas para classificar cada pixel como pele ou não-pele.

3.4.1.1 – Modelo RGB

O vector cor é representado pelas três componentes do modelo RGB:

$$C = [R, G, B] \quad (3.7)$$

Através da aplicação das seguintes regras [17] usadas por Peer a todos os pixels da imagem:

Quando $R < 95$ e $G > 40$ e $B > 20$ e $\max\{R,G,B\} - \min\{R,G,B\} > 15$ e $|R - G| > 15$ e $R > G$ e $R > B$, é classificado como um pixel com pele.

É obtida uma imagem binária (valores 0 e 1), onde a cor branca representa as zonas de pele da imagem e a cor preta representa as zonas de não-pele da imagem.

3.4.1.2 – Modelo HSV

O vector cor é representado por duas componentes do modelo HSV:

$$C = [H, S] \quad (3.8)$$

Após a análise de 10 amostras de pele extraídas com o programa Microsoft Picture Manager de 16x16 pixels de pessoas de etnia branca, foi possível verificar que os valores de cada pixel da pele podem ser definidos segundo regras. As regras são as seguintes:

Quando $(S < 0.85$ e $S > 0.2)$ e $(H > 0$ e $H < 0.15)$ é classificado como um pixel de pele.

É obtida uma imagem binária onde a cor branca representa as zonas de pele da imagem e a cor preta representa as zonas de não-pele da imagem.

3.4.2 – Classificador Gaussiano

Neste método [19] são usadas as duas componentes de crominância dos espaço de cores YCrCb, separando a componente da luminosidade:

$$C = [Cr \ Cb] \quad (3.9)$$

A condição para determinar a probabilidade de cada pixel cor ser pele é obtida através da função densidade de probabilidade conjunta Gaussiana, que é a seguinte:

$$P(C / Pele) = \frac{1}{2\pi |\Sigma|^{1/2}} e^{\frac{1}{2}(C-\mu)^T \Sigma^{-1}(C-\mu)} \quad (3.13)$$

onde C é a cor do pixel a classificar, μ e Σ são os parâmetros de distribuição, μ é o vector media (Cr e Cb) e Σ é a matriz covariância, valores obtidos através do conjunto de treino:

$$\mu_i = \frac{1}{n} \sum_{i=1}^n c_i \quad (3.14)$$

$$\Sigma = \frac{1}{n-1} \sum_{j=1}^n (c_j - \mu_s)(c_j - \mu_s)^T \quad (3.15)$$

Onde n é o numero de amostras de cor da pele presentes no conjunto de treino.

Na figura 3.5 está representada a distribuição Gaussiana para os valores da crominância obtidos no conjunto de treino.

O conjunto de treino é constituído por 90 amostras de pele de tamanho 16x16 extraídas manualmente com o programa Microsoft Picture Manager das imagens da base de dados [22]. As amostras de pele são de pessoas de ambos os sexos e etnia branca.

Após a obtenção da probabilidade de $P(C/Pele)$, define-se um limiar (pode ser definido empiricamente ou pode ser calculado computacionalmente). Caso a $P(C/Pele)$ seja superior ao limiar, o pixel é pele, caso contrario o pixel é não-pele, consequentemente é obtida uma imagem binária onde a cor branca representa a pele e a cor preta representa não-pele.

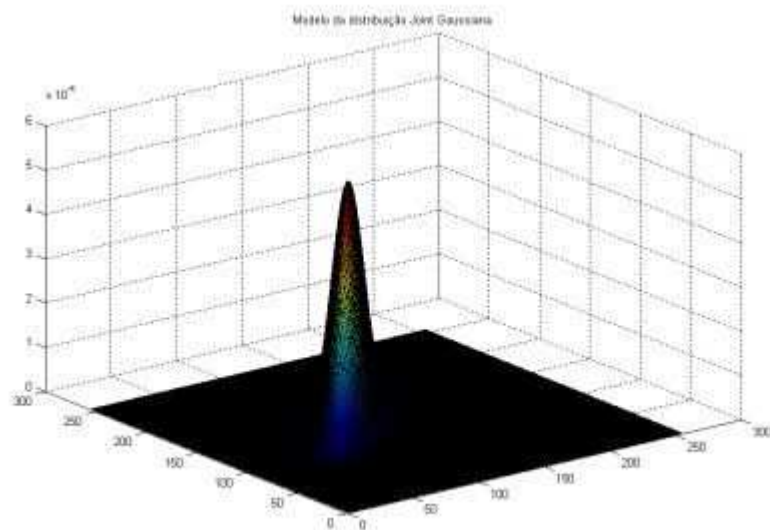


Figura 3.5 – Modelo de distribuição Joint Gaussiana para o conjunto de treino

É possível obter um modelo com mistura de Gaussianas, informação detalhada sobre o método em [13].

3.5 – Operadores morfológicos em imagens binárias

A imagem binária obtida anteriormente contém todas as faces da imagem, todas as partes de pele visíveis do corpo e algumas partes do fundo. De forma a reduzir as falsas detecções de fundos e eliminar os buracos dentro das faces aplicam-se operadores morfológicos binários.

3.5.1 – Erosão e dilatação

Os operadores morfológicos aplicados neste trabalho são a erosão e a dilatação.

Primeiro é aplicado à imagem uma erosão de 9x9 pixels de forma a eliminar as pequenas regiões de fundo mal detectadas e separar as regiões a classificar. Posteriormente a imagem erodida é dilatada de 10x10 pixels de forma a encher novamente os buracos nas faces.

Estas duas operações complementam-se. A primeira serve para eliminar detalhes irrelevantes de uma figura a partir da redução das figuras que compõem uma imagem usando um elemento estruturante, elimina todas as regiões que contenham uma área até 81 pixels.

A segunda serve para eliminar buracos na imagem (face) e aumenta o tamanho da imagem. Em suma, é feita uma correcção (diminuição do erro), isto é, vai haver um preenchimento de pequenos buracos (passa um pixel isolado de 0 para 1) e elimina regiões isoladas que contenham poucos pixels (passa uma pequena zona de pixels de 1 para 0). A erosão matematicamente pode ser descrita [16]:

$$A \ominus B = B \cap A = \{x | (B)_x \subseteq A\} \quad (3.16)$$

Sendo B o elemento estrutural, que neste caso é o quadrado, A a imagem binária. Simplificadamente é possível definir o resultado da erosão binária da imagem A pelo elemento estruturante B como todas as localizações X do elemento estruturante B na imagem A.

A dilatação matematicamente pode ser descrita [16]:

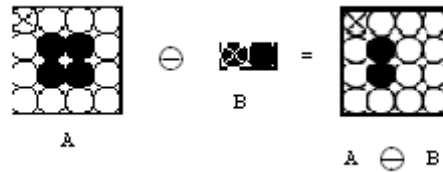


Figura 3.6 – Aplicação da erosão para um conjunto de pixels

$$A \oplus B = A \cup B = \{x | [(B)_x \cap A] \neq \emptyset\} \quad (3.17)$$

Sendo B o elemento estrutural, que neste caso é o quadrado, A a imagem binária. Pode ser definida como a posição dos pontos x, de forma a que o elemento estruturante B coincide com A quando a sua origem está centrada em x. Simplificadamente é possível definir o resultado da dilatação binária da imagem A pelo elemento estruturante B como todas as localizações do elemento estruturante B na imagem A, onde pelo menos um pixel coincida.

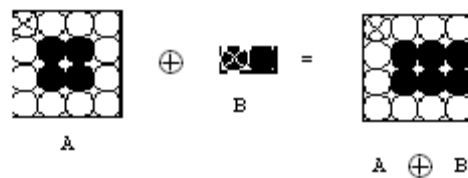


Figura 3.7 – Aplicação da dilatação para um conjunto de pixels

Após processar o método da erosão e da dilatação são obtidas as diferentes regiões de pele para posteriormente serem analisadas

separadamente. Consequentemente é possível obter as diferentes características de cada região (área, centro, “boundingbox” e orientação).

Um artigo que explica detalhadamente todo o processo de erosão e dilatação em imagens binárias pode ser consultado em [16].

3.6 – Método do Protótipo

Após a obtenção das diferentes regiões de pele e de algum fundo mal detectado é necessário desenvolver um algoritmo que detecte se as diferentes regiões a classificar são face ou não face. Em caso afirmativo determinar as coordenadas da região e desenhar um rectângulo que contenha o conjunto olhos, boca e nariz.

O método do protótipo [4] [9] consiste na referência de um modelo protótipo que foi construído manualmente ou estereotipado no conjunto de treino, que depois de ajustado na imagem a classificar, e através da correlação cruzada permite obter a semelhança existente entre o modelo e a região a classificar.

3.6.1 – Protótipo

O protótipo usado foi uma elipse binária construída manualmente. Onde a parte de dentro da elipse está representada a branco e representa pixels que correspondem à face e a parte de fora da elipse representa pixels que correspondem a não-face. Foi escolhida uma elipse porque é

uma figura geométrica que se assemelha à face humana média. O tamanho original do template é 53X81 pixels. Na figura 3.8 é possível observar o protótipo usado para comparar com a região a classificar.



Figura 3.8 – Protótipo implementado

3.6.2 - Correlação cruzada

Anteriormente à aplicação do método da correlação cruzada é necessário determinar o ângulo que o rectângulo da região a classificar faz com o rectângulo da imagem. Como se pode observar na figura seguinte, a região a classificar não está no mesmo eixo de orientação que apresenta na imagem, logo é necessário determinar o ângulo θ para poder determinar o eixo de orientação da região a classificar.

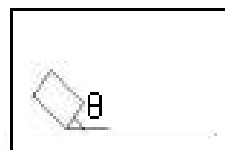


Figura 3.9 – Ângulo da região a classificar

Posteriormente é necessário obter as coordenadas dos quatros vértices do rectângulo da região a classificar na imagem. A área que delimita a região a classificar é obtida através da extracção do vértice inferior esquerdo do rectângulo, do comprimento e da largura da região a classificar. Com o vértice esquerdo, a largura, o comprimento e o ângulo que a região a classificar faz com a imagem são obtidos os restantes vértices que delimitam o rectângulo.

È muito importante adaptar convenientemente o protótipo em relação à região a classificar de forma a obter bons resultados. O protótipo é ajustado e é colocado na exacta posição da região a comparar na imagem.

Em seguida é efectuado um pré-processamento através de técnicas baseadas no conhecimento das características da face.

Tendo em conta os conhecimentos sobre as dimensões da face humana, é possível eliminar algumas das regiões a classificar. Efectuada uma análise a todas as regiões, é fácil identificar que quando a dimensão de comprimento é bastante diferente da dimensão da largura, a região a classificar não corresponde a uma face, dado que a face media humana não difere muito em comprimento e largura. Se:

$$0.5 < \frac{\textit{largura}}{\textit{comprimento}} < 2 \quad (3.18)$$

a região a classificar é uma região que possivelmente contém uma face, caso não esteja dentro desse intervalo a região é uma não-face.

Outra grande dificuldade é a separação da face com o pescoço. Quando é feita a detecção das zonas de pele, e caso o pescoço não esteja tapado por uma camisola ou outro objecto, a região a classificar vai apresentar a cara e o pescoço. Para eliminar o pescoço é feita uma comparação entre a largura e o comprimento.

$$\text{Se } \frac{\text{largura}}{\text{comprimento}} > 1.25, \text{ largura} = \text{largura} * 0.8 \quad (3.19)$$

$$\text{Se } \frac{\text{largura}}{\text{comprimento}} < 0.8, \text{ comprimento} = \text{comprimento} * 0.8 \quad (3.20)$$

É necessário ter em conta que a face não aparece sempre na imagem com a mesma orientação. Observando a figura 3.10, é possível verificar que o pescoço aparece em sítios diferentes se a orientação da imagem é diferente. Tendo em conta a orientação da imagem é necessário ou reduzir a largura ou o comprimento, como é possível verificar nas imagens. Analisando as imagens 3.11 e 3.12 é muito fácil verificar que é necessário reduzir o comprimento na imagem da esquerda e reduzir a largura na imagem da direita de forma a detectar a face sem a parte do pescoço.

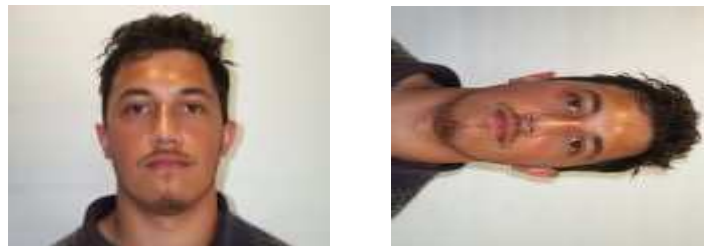


Figura 3.10 – Imagem original e imagem original rodada



Figura 3.11 – Zonas de pele detectadas



Figura 3.12 – Zonas de pele detectadas delimitadas por um rectângulo

Esta diminuição do comprimento ou da largura permite eliminar o pescoço da região a classificar.

A correlação cruzada é um método [4] [7] que consiste no uso de um template que percorre toda a imagem ou região a classificar, e permite a obtenção de informações de semelhança com a região a classificar. Anteriormente a aplicar o método da correlação cruzada é necessário alterar o template de forma a obter o mesmo ângulo em relação à imagem e a mesma dimensão que a região a classificar. O método da correlação é aplicado entre o template e a região a classificar e é obtida a semelhança entre ambas.

Matematicamente o coeficiente de correlação [32] pode ser descrito pela seguinte equação:

$$CC = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (f(m,n) - \mu_f)(g(m,n) - \mu_g)}{\sqrt{(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (f(m,n) - \mu_f)^2)(\sum_{m=0}^{M-1} (g(m,n) - \mu_g)^2)}} \quad (3.21)$$

Onde m corresponde ao número da linha na matriz, n corresponde ao número da coluna na matriz, μ_f representa a média de todos os valores dentro da matriz f e μ_g representa a média de todos os valores dentro da matriz g .

Foi definido que se a correlação for superior a 63% a região a classificar é uma face. Nesse caso é desenhado um rectângulo na face.

4 – Resultados

4.1 – Bases de dados

O algoritmo é testado em duas bases de dados públicas, a primeira base de dados foi generosamente cedida por Meredith Minear e Denise C. Parker [21]. É uma base de dados que contém imagens de tamanho 640x480 com uma face e fundo simples. Possui faces com diferentes sexos, diferentes etnias e faixa etária desde os 18 anos até aos 94 anos. Contém também imagens de perfil e de pessoas com sete estados emocionais que não foram usadas para testar o algoritmo. A segunda base de dados testada foi obtida foi criada por Markus Weber [22], possui 450 imagens de tamanho 896x582 com uma face, de 27 pessoas com diferentes iluminações, expressões faciais e fundos. Todas as imagens foram reduzidas manualmente no programa Microsoft Office Picture Manager para o formato de “mensagem electrónica” de forma a economizar tempo durante os testes.

4.2 – Interface

Na figura 4.1 é possível observar o menu do programa.



Figura 4.1 – Menu do programa

No capítulo 8 (Apêndice) é possível consultar as principais funções do algoritmo desenvolvido.

4.3 – Detecção de faces

São apresentados os resultados de duas imagens, uma imagem com uma face com um fundo simples e a outra com um fundo complexo, como é possível verificar na figura 4.2.

4.3.1 – Classificador baseado em regras (RGB)

As imagens originais são as seguintes:



Figura 4.2 – Imagens originais

Imagens binárias, onde branco significa as zonas de pele detectadas pelo algoritmo e o preto as zonas de não pele detectadas pelo algoritmo:



Figura 4.3 – Imagens binárias com as zonas de pele detectadas

Imagens binárias depois de aplicar o método da erosão e da dilatação:



Figura 4.4 – Imagens binárias depois do método da erosão e dilatação

Neste caso como foram detectadas duas regiões de pele, é necessário separar ambas as regiões para aplicar posteriormente e separadamente a classificação das diferentes regiões de pele.



Figura 4.5 – As duas regiões de pele separadas

Imagens finais com a detecção e localização da face na imagem, já depois da aplicação das considerações geométricas e do método do template:



Figura 4.6 – Imagens finais com a detecção

4.3.2 – Classificador baseado em regras (HSV)

Para o espaço de cores HSV são apresentados os resultados para as mesmas imagens, as imagens binárias com as detecções das zonas de pele:



Figura 4.7 – Imagens binárias com as zonas de pele detectadas

Imagens binárias depois de aplicar o método da erosão e da dilatação:



Figura 4.8 – Imagens binárias depois do método da erosão e dilatação

Imagens finais com a detecção e localização da face na imagem, já depois da aplicação das considerações geométricas e do método do template:

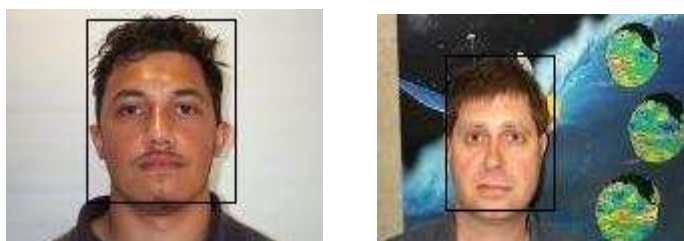


Figura 4.9 – Imagens finais com a detecção

4.3.3 – Classificador Gaussiano

Para o espaço de cores YCrCb são apresentados os resultados para as mesmas imagens, as imagens binárias com as detecções das zonas de pele:



Figura 4.10 – Imagens binárias com as zonas de pele detectadas

Imagens binárias depois de aplicar o método da erosão e da dilatação:



Figura 4.11 – Imagens binárias depois do método da erosão e dilatação

Imagens finais com a detecção e localização da face na imagem, já depois da aplicação das considerações geométricas e do método do template:

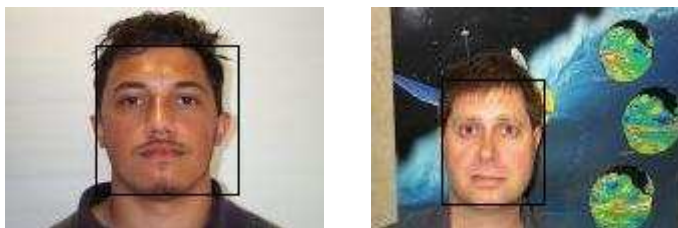


Figura 4.12 – Imagens finais com a detecção

4.4 – Comparação dos detectores de pele

De forma a determinar qual o detector de pele que funciona melhor foram efectuadas 30 detecções de face em imagens com fundo simples. Em baixo são apresentadas as taxas de acerto na detecção e localização da face na imagem para cada um dos métodos:

RGB	YCrCb	HSV
29/30	23/30	26/30

Tabela 4.1 – Resultados dos detectores de pele

RGB	YCrCb	HSV
96,67%	76,67%	86,67%

Tabela 4.2 – Resultados dos detectores de pele (%)

Após a observação dos resultados e análise detalhada dos resultados de ambos os detectores, apenas se utilizou o método com o espaço de cores RGB nos testes seguintes.

4.5 – Imagens de face com fundo simples

Esta base de dados de imagens de face com fundo simples possui imagens de sexos distintos, etnias distintas e idades compreendidas entre os 18 e os 94 anos.

Exemplos de detecções bem sucedidas para imagens com diferentes sexos, etnias e idades:



Figura 4.13 Face masculina, idade 19, etnia branca

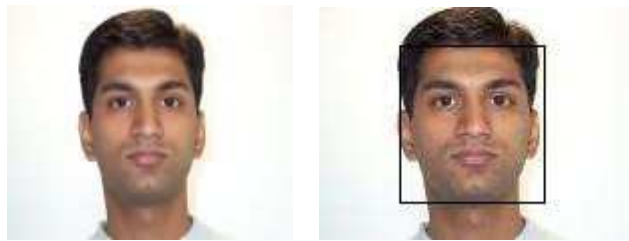


Figura 4.14 Face masculina, idade 24, etnia índio



Figura 4.15 Face feminina, idade 25, etnia india



Figura 4.16 Face feminina, idade 37, etnia negra

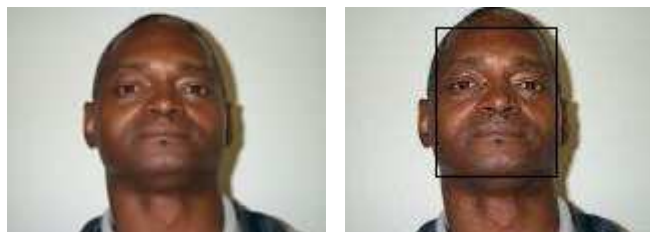


Figura 4.17 Face masculina, idade 50, etnia negra



Figura 4.18 Face feminina, idade 70, etnia branca

Resultados obtidos para imagens com fundo simples, diferentes sexos, etnias e idades:

Sexo(faixa etária)	Correctas/totais
M(18-29)	107/114
F(18-29)	89/106
M(30-49)	26/29
F(30-49)	40/46
M(50-69)	28/29
F(50-69)	81/89
M(70-94)	43/46
F(70-94)	96/108

Tabela 4.3 – Resultados para imagens com faces de diferentes sexos e idades

Sexo(faixa etária)	Porcentagem
M(18-29)	93,86%
F(18-29)	83,96%
M(30-49)	89,66%
F(30-49)	86,96%
M(50-69)	96,55%
F(50-69)	91,01%
M(70-94)	93,48%
F(70-94)	88,89%

Tabela 4.4 – Resultados para imagens com faces de diferentes sexos e idades (%)

Sexo	Correctas/Totais
Masculino	204/218
Feminino	307/349

Tabela 4.5 – Resultados para imagens com diferentes sexos

Sexo	Percentagem
Masculino	93,58%
Feminino	87,96%

Tabela 4.6 – Resultados para imagens com diferentes sexos (%)

faixa etária	Correctas/Totais
18-29	196/220
30-49	66/75
50-69	110/118
70-94	139/154
Total	511/567

Tabela 4.7 – Resultados para imagens com diferentes idades

faixa etária	Percentagem
18-29	89,09%
30-49	88%
50-69	93,22%
70-94	90,26%
Total	90,12%

Tabela 4.8 – Resultados para imagens com diferentes idades (%)

Nesta base de dados foram obtidos 4 falsos positivos.

4.6 – Imagens de face com fundo complexo

Esta base de dados possui imagens com distintos sexos e etnia branca com uma face e com fundos complexos e com diferentes iluminações, em 350 imagens foi detectada e localizada a face 297 vezes, o que significa uma taxa de acerto de 84,86%. Foram obtidos também 12 falsos positivos. Esta base de dados foi também testada usando o demo explicado no artigo[10], extraído do site [11]. Os resultados obtidos com o algoritmo desenvolvido por Wolf [10] obtiveram 320 faces detectadas correctamente em 350 imagens, o que significa uma taxa de acerto de 91,43%. Foram obtidos 28 falsos positivos.

Alguns exemplos das detecções correctas:



Figura 4.19 – Imagem original e imagem final com a detecção



Figura 4.20 - Imagem original e imagem final com a detecção



Figura 4.21 – Imagem original e imagem final com a detecção



Figura 4.22 – Imagem original e imagem final com a detecção

4.7 – Imagens de grupo

Alguns exemplos do algoritmo aplicado a imagens de grupo:



Figura 4.23 – Imagem final com a detecção



Figura 4.24 – Imagem final com a detecção



Figura 4.25 – Imagem final com a detecção

5 - Conclusões

O algoritmo detecta e localiza faces humanas em imagens de cor com diferentes etnias, idades compreendidas entre os 18 e 94 anos, diferentes sexos, diferentes fundos, com luminosidade e diferentes expressões faciais.

Foram testados 3 detectores de pele, após análise detalhada dos resultados foi possível concluir que o melhor detector de pele implementado foi o método com a região de pele definida no espaço de cores RGB. Este apresentou 96,67% de acerto nas 30 imagens testadas. Apresentou também visivelmente um maior acerto nos pixels que representam a pele. O método de região definida com o espaço de cores HSV apresentou um acerto de 86,67% e o método da distribuição gaussiana com o espaço de cores YCrCB obteve um acerto de 76,67%. O resultado menos conseguido do método da distribuição gaussiana deve-se possivelmente às inadequadas amostras do conjunto de treino. As duas bases de dados foram testadas com o detector de pele implementado com a região de pele definida no espaço de cores RGB.

O resultado obtido para a base de dados [21] foi 90,12% (511/567) de acerto e 4 falsos positivos. Analisando outros trabalhos é possível verificar taxas de acerto desde os 77,47% [33] até aos 96,01% [34] para imagens coloridas de uma única face, testadas em base de dados de domínio público.

Através da análise dos resultados para imagens de face com ambos os sexos e com fundo simples, é possível concluir que o algoritmo funciona ligeiramente melhor para o sexo masculino (93,58% (204/218)) do que para o feminino (87,96% (307/349)). É um resultado perfeitamente previsível, porque o sexo feminino na generalidade apresentam o cabelo mais longo, o que por vezes resulta numa oclusão parcial da cara, e consequentemente o algoritmo apresenta uma ligeira diferença relativamente ao sexo masculino.

Relativamente à faixa etária é possível concluir que a faixa etária não influencia na eficiência do algoritmo. O algoritmo apresenta resultados semelhantes nas diferentes faixas etárias definidas. Os resultados obtidos foram para a faixa etária dos 18 aos 29 anos 89,09% (196/220) de acerto, dos 30 aos 49 anos 88% (66/75) de acerto, dos 50 aos 69 anos 93,22% (110/118) de acerto e dos 70 aos 94 anos 90,26% (139/154) de acerto.

Como é demonstrado nos resultados o algoritmo funciona para diferente etnias, foram testadas 3 etnias (branco, negro e índio). Não foi possível apresentar conclusões sobre se a etnia influencia na eficiência do algoritmo devido a escassez de imagens da etnia negra e índia nas bases de dados testadas.

Os testes efectuados na base de dados [22] em imagens com diferentes sexos, fundos, expressões faciais e iluminações resultaram em uma eficiência de 84,86% (297/350) e 12 falsos positivos. A mesma base de dados foi testada em um demo [11] obtendo uma eficiência de 91,43% (320/350) e 28 falsos positivos. Analisando ambos os resultados é possível concluir que a diferença de resultados deve-se em grande parte à fraca eficiência apresentada pelo algoritmo implementado neste trabalho em imagens com presença de luminosidade e também às imagens que o fundo se assemelha muito à cor de pele. Em relação aos falsos positivos é possível concluir que o algoritmo implementado neste trabalho apresenta uma detecção de falsos positivos muito abaixo do demo extraído. Se analisar os resultados obtidos na base de dados, onde não há presença de luminosidade e os fundos são simples. É possível verificar que obteve uma eficiência de 90,12%, conseqüentemente é possível concluir que o algoritmo implementado apresenta uma menor eficiência em imagens com luminosidade.

Os resultados permitem concluir que o algoritmo funciona melhor para a primeira base de dados [21] testada que para a segunda [22].

Segundo Veznhets [13] o espaço de cores RGB não é uma escolha favorável para a análise da cor e para algoritmos de reconhecimento facial,

como foi referido no ponto 3.3.1. Contraditoriamente ao referido, os resultados obtidos permitem concluir que o algoritmo implementado funciona eficientemente para as duas bases testadas.

O algoritmo foi testado em algumas imagens de grupo com sucesso, como é possível verificar no ponto 4.7. O algoritmo tem algumas dificuldades em detectar todas as faces numa imagem de grupo quando as faces não possuem o mesmo tamanho que o template ou quando na mesma imagem se encontram faces com diferentes escalas.

Observando a imagem 4.23 é possível verificar que quando o cabelo é loiro muito claro o detector de pele considera o cabelo como pele e a detecção da face inclui também o cabelo.

O algoritmo não funciona para imagens com face de perfil, como seria previsível, dado que o algoritmo não foi implementado para faces na posição de perfil.

Para finalizar é possível concluir que os resultados obtidos foram bastante satisfatórios, tendo em conta que é um trabalho inicial no âmbito da detecção e localização de faces em imagens.

6 – Trabalho futuro

Após a conclusão do trabalho e identificação das limitações do sistema proponho a implementação de uma função para resolver o problema da luminosidade eficazmente.

Efectuar extracções de características faciais de forma a melhorar o algoritmo.

Construir uma função que detecte faces na posição em perfil.

Implementar outro detector de pele, por exemplo, a mistura de Gaussianas, de forma a melhorar o detector de pele.

Implementar outro detector de faces para comparar resultados, por exemplo, implementar o método do PCA.

7 - Bibliografia

- [1] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, Jan. 1998, pgs. 23-38.
- [2] H. Rowley, "Neural Network-Based Face Detection," Ph.D. Thesis Carnegie Mellon University, CMU-CS-99-117, 1999.
- [3] R-L. Hsu, and M. Abdel-Mottaleb, "Face Detection in Color Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, May 2002, pgs. 696 - 706.
- [4] Fernando J. S. Carvalho and João M. R. S. Tavares, "Metodologias para identificação de faces em imagens: introdução e exemplos de resultados," Congresso de Métodos Numéricos en Ingeniería 2005, Granada, España, 4 – 7 de julio de 2005,.
- [5] M. H. Yang, and D. J. Kriegman, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, Jan. 2002, pgs. 34 – 58.
- [6] E. Hjelmås, and B. K. Low, "Face Detection: A Survey, *Computer Vision and Image Understanding*," Vol. 83, No. 3, Sept. 2001, pgs. 236-274.
- [7] R. Brunelli and T. Poggio, "Face Recognition: Features Versus Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, Oct. 1993, pgs. 1,042–1,052.
- [8] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," presented at IEEE Conference on Computer Vision and Pattern Recognition, 1999.
- [9] A. L. Yuille, D. S. Cohen, and P.W. Hallinan, "Feature extraction from faces using deformable templates," *Proc.CVPR*, 1989, pgs. 104–109.
- [10] W. Kienzle, G. Bakir, M. Franz and B. Scholkopf, "Face Detection - Efficient and Rank Deficient," *Advances in Neural Information Processing Systems* 17, 2005, pgs. 673-680.
- [11] W. Kienzle, G. Bakir, M. Franz and B. Scholkopf, "Fdlib," <http://www.kyb.mpg.de/bs/people/kienzle/fdlib/fdlib.htm>

- [12] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer.* 4, 1987, pgs. 519–524.
- [13] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," *Graphicon-2003.*, pgs.85–92, Moscow, 2003
- [14] Zarit, B. D., Super, B. J., and Quek, "Comparison of five color models in skin pixel classification," *ICCV'99 Int'l Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems*, pgs 58-63.
- [15] Terrillon, J.-C., Shirazi, M. N., Fukamachi, H., and Akamatsu, S. "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images," *International Conference on Face and Gesture Recognition*, 2000, pgs 54-61.
- [16] H. Hedberg, F. Kristensen and V. Öwall, "Low-Complexity Binary Morphology Architectures with Flat Rectangular Structure Elements," *IEEE International Symposium on Circuits and Systems*, Vol. 4, May 2005, pgs. 3431-3434.
- [17] PEER, P., KOVAC, J., AND SOLINA, F., "Human skin colour clustering for face detection," *EUROCON 2003 – International Conference on Computer as a Tool, 2003.*
- [18] ALBIOL, A.; TORRES, L.; DELP, E.J. "Optimum Color Spaces for Skin Detection," *IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001, pgs 122–124.
- [19] Caetano T S, Olabarriaga S D, Barone D A C, "Performance Evaluation of Single and Multiple-Gaussian Models for Skin Color Modeling, *Brazilian Symposium on Computer Graphics and Image Processing*," XV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAP). Washington, DC, USA: IEEE Computer Society, 2002, pgs 275-282
- [20] P. Soille, "Morphological image analysis applied to crop field mapping," *Image and Vision Computing*, Vol. 18, No. 13, 1998, pgs. 1025-1032.

- [21] Minear, M. & Park, D.C., "A lifespan database of adult facial stimuli," Behavior Research Methods, Instruments, & Computers, <https://pal.utdallas.edu/facedb>.
- [22] Weber, Markus, "Faces 1999 (Front)," The Computational Vision Group, <http://www.vision.caltech.edu/html-files/archive.html>.
- [23] GONZALEZ, R.C.; WINTZ, "P. *Digital Image Processing Reading*," MA. Addison, 1987.
- [24] Yang Jie, Lu Weier, and Waibel Alex. "Skin-color modeling and adaptation," Computer Vision - ACCV'98, Universitat Karlsruhe; Institut fur Logik, Komplexit at und Deduktionssysteme, R. Chin. Vol. 2., 1997, pgs 687-694.
- [25] Poynton S, Fraser W, Francis-Floyd R, Rutledge P, Reed P, Nerad T, "Spironucleus vortens n. sp. from the freshwater angelfish *Pterophyllum scalare*: morphology and Culture," 1995, J Eukaryot Microbiol, pgs 731-742
- [26] J. Foley, A. van Dam, S. Feiner, J. Hughes; "Computer Graphics: Principles and Practice," Addison-Wesley; 1996
- [27] JONES, M. J., AND REHG, J. M., "Statistical color models with application to skin detection", CVPR '99, vol. 1, 1999, pgs 274-280
- [28] G. Yang and T. S. Huang, "Human Face Detection in Complex Background," Pattern Recognition, vol. 27, no. 1, 1994, pgs. 53-63.
- [29] S. McKenna, S. Gong, and Y. Raja, "Modelling Facial Colour and Identity with Gaussian Mixtures," Pattern Recognition, vol. 31, no. 12, 1998, pgs. 1883-1892.
- [30] I. Craw, D. Tock, and A. Bennett, "Finding Face Features," Second European Conf. Computer Vision, 1992, pgs. 92-96.
- [31] A. Samal and P. A. Iyengar, "Human face detection using silhouettes," *Int. J. Pattern Recog. Artificial Intell.* 9(6), 1995
- [32] Gonzalez, R.C. and Woods, R.E., "Digital image processing, 2nd ed.," Pearson Education, Inc, publishing as Prentice Hall, 2002.

- [33] Moutinho, Adriano M.; Thomé, Antonio Carlos Gay, “Localização de faces em imagens coloridas usando redes neurais,” Universidade Federal do Rio de Janeiro, núcleo de Eletrônica, 2006
- [34] S.K. Singh, D.S. Chauhan, M. Vatsa and R. Singh, “A robust skin color based face detection algorithm,” *Tamkana Journal of Science and Engineering*, vol.g, no.4, 2004, pgs. 227-234.
- [35] D. Chai and K.N. Ngan, “Locating Facial Region of a Head-and-Shoulders Color Image,” Third Int Conf. Automatic Face and Gesture Recognition, pgs. 124-129, 1998.
- [36] T. Sakai, M. Nagao, and S. Fujibayashi, “Line Extraction and Pattern Detection in a Photograph,” *Pattern Recognition*, vol. 1, pgs. 1969, 233-248.
- [37]Md. A.A. Buihyan, V.Ampornaramveth,S.Yo H. Ueno, “Face Detection and Facial Feature Localization for Human-machine Interface,” *NII Journal No.5, 2003.*
- [38]Sung,K.-K, Poggio T, “Example based learning for view based human facedetection,” *Pattern Analysis and Machine Intelligence*, IEEE transactions on,volume 0,Issue 1 Jan 1998, pgs. 39 – 51.
- [39]R.M. Haralick, K. Shanmugam, and I. Dinstein, “Texture Features for Image Classification,” *IEEE Trans. Systems, Man, and Cybernetics*, vol. 3, no. 6, 1973, pgs. 610-621.
- [40]I.A. Essa and A. Pentland, “Facial Expression Recognition Using a Dynamic Model and Motion Energy,” Fifth IEEE Int’l Conf. Computer Vision, 1995, pgs. 360-367.
- [41]S. Fahlman and C. Lebiere, “The Cascade-Correlation Learning Architecture,” *Advances in Neural Information Processing Systems 2*, D.S. Touretsky, ed., 1990, pgs. 524-532.

8. Apêndice

Em seguida são apresentadas algumas das principais funções usadas para construção do algoritmo.

```
%Interface inicial com todas as opções
```

```
clear all;
```

```
close all;
```

```
[nome_ficheiro caminho_ficheiro] = uigetfile (*.jpg);
```

```
    if caminho_ficheiro ~= 0
```

```
        imagem0 = imread ([caminho_ficheiro,nome_ficheiro]);
```

```
    end
```

```
imshow(imagem0)
```

```
pause()
```

```
    escolha = menu('Deteccao Faces',...
```

```
        'Espaço RGB - Metodo Template',...
```

```
        'Espaço YCrCb - Metodo Template',...
```

```
        'Espaço HSV - Metodo Template',...
```

```
        'Sair');
```

```
    if ( escolha == 1)
```

```
        imagem_pele_binaria = pele_RGB(imagem0);
```

```
        imagem_pele_final = metodo_ero_dil(imagem_pele_binaria);
```

```
        [imagem_pele_template,dados]=metodo_template2(imagem0,imagem_pele_final);
```

```
        desenha_rectangulos(imagem_pele_template,dados,imagem0);
```

```
    end
```

```
    if ( escolha == 2)
```

```

imagem_pele_binaria=pele_vero(imagem0);
imagem_pele_final = metodo_ero_dil(imagem_pele_binaria);
[d,dados]=metodo_template2(imagem0,imagem_pele_final);
desenha_rectangulos(imagem_pele_template,dados,imagem0);
end

```

```

if (escolha == 3)

```

```

    imagem_pele_binaria = pele_HSV2(imagem0);
    imagem_pele_final = metodo_ero_dil(imagem_pele_binaria);
    [d,dados]=metodo_template2(imagem0,imagem_pele_final);
    desenha_rectangulos(imagem_pele_template,dados,imagem0);
end

```

```

if (escolha == 4)

```

```

    clear all;
    clc;
    close all;
    opcao = 0;
end

```

%Função que recebe como parâmetro de entrada a imagem original e devolve como
%parâmetro de saída uma imagem binária com as zonas de pele detectadas

```

function imagem_pele_binaria_RGB = pele_RGB(imagem_lum)

```

```

imagem=imagem_lum;

```

```

R=imagem(:,:,1);

```

```

G=imagem(:,:,2);

```

```

B=imagem(:,:,3);

```

```

[x,y]=size(R);

```

```

% Aplicação das regras a cada pixel
%Se o pixel corresponde a todas as regras é 1 (pele)
%Se não corresponde é 0(não-Pele)
for l=1:x
    for c = 1:y
        pixel_R = R(l,c);
        pixel_G = G(l,c);
        pixel_B = B(l,c);

        max_RGB = max(max(pixel_R, pixel_G), pixel_B);
        min_RGB = min(min(pixel_R, pixel_G), pixel_B);
        condicao_a = (pixel_R>95 && pixel_G>40 && pixel_B>20);
        condicao_b = max_RGB-min_RGB>15 ;
        condicao_c = abs(pixel_R-pixel_G)>15;
        condicao_d = pixel_R>pixel_G & pixel_R>pixel_B;
        if (condicao_a && condicao_b & condicao_c && condicao_d)
            imagem_pele_binaria_RGB(l,c)=1;

        else

            imagem_pele_binaria_RGB(l,c)=0;

        end
    end
end

imshow(imagem_pele_binaria_RGB)
pause()

```

```

%função que aplica o método da erosão e da dilatação
%à imagem binária

```

```

function [imagem_pele_final] = metodo_ero_dil(imagem_pele_binaria)

[linha,coluna] = size(imagem_pele_binaria);

%preenche as regiões com pequenos buracos
preenche_regioes = zeros(linha,coluna);
preenche_regioes = imfill(imagem_pele_binaria,'holes');

imshow(imagem_pele_binaria)
pause()

%Aplica a erosão e a dilatação

dilatacao=zeros(linha,coluna);
dilatacao=imdilate(imerode(preenche_regioes,strel('square',9)),strel('square',10));
%Multiplica a imagem erodida com a imagem segmentada

dilatacao = immultiply(dilatacao,imagem_pele_binaria);

imshow(dilatacao)
title('Imagem depois da dilatacao')

pause()

%Marca as regiões de pele
marca_regiao=zeros(linha,coluna);
[marca_regiao,num] = bwlabel(dilatacao,8);

imshow(marca_regiao);
title('Grafico com as regioes de peles separadas');
pause()

area0 = regionprops(marca_regiao,'Area');

area=cat(1,area0.Area);

numero = find(area>100)

```

```

[linha,coluna]=size(marca_regiao);

imagem_pele_final=zeros(linha,coluna);

numero_regioes=length(numero);

i=1;

%imagem binária que apenas com regiões de pele segmentadas
while (i < numero_regioes+1)

    % Obtenção das coordenadas dos pixels de cada região
    [x,y] = find(bwlabel(marca_regiao) == numero(i));

    % imagem apenas com as regiões de interesse, o resto a preto
    imagem_segmentada_nova = bwselect(marca_regiao,y,x,8);

    imagem_pele_final=imagem_pele_final+imagem_segmentada_nova;
    i=i+1;
end

%imagem binária com as regiões delimitadas e etiquetas para os elementos conectados

%preenche os buracos nas possíveis regiões a classificar
preenche = imfill(imagem_pele_final,'holes');
[linha,coluna]=size(imagem_pele_final);
imagem_pele_preenchida=zeros(linha,coluna);
imagem_pele_preenchida=imdilate(preenche,strel('square',2));
imagem_pele_final=imagem_pele_preenchida;

%  imshow(imagem_pele_final)

% pause()

%Função que determina o ângulo para cada região

function angulos = determina_angulos(imagem_pele_final,etiquetas)

```

```
%Angulo para cada regioao
```

```
orient = regionprops(etiquetas,'Orientation');
```

```
angulos=cat(1,orient.Orientation);
```

```
% Esta função modifica as coordenada da imagem,  
%para o limite da imagem, caso estas estejam forma  
%da imagem
```

```
function [inicioX, inicioY, fimX,fimY] = limites(inicioX,inicioY,fimX,fimY,limX,limY)
```

```
nova_cordenada_inicio = inicioX;
```

```
if (nova_cordenada_inicio<1)
```

```
    nova_cordenada_inicio=1;
```

```
end
```

```
if (nova_cordenada_inicio>limX)
```

```
    nova_cordenada_inicio=limX;
```

```
end
```

```
inicioX=nova_cordenada_inicio;
```

```
nova_cordenada2_inicio = inicioY;
```

```
if (nova_cordenada2_inicio<1)
```

```
    nova_cordenada2_inicio=1;
```

```
end
```

```
if (nova_cordenada2_inicio>limY)
```

```
    nova_cordenada2_inicio=limY;
```

```
end
```

```
inicioY = nova_cordenada2_inicio
```

```
nova_cordenada_fim = fimX;
```

```
if (nova_cordenada_fim<1)  
    nova_cordenada_fim=1;  
end
```

```
if (nova_cordenada_fim>limX)  
    nova_cordenada_fim=limX;  
end
```

```
fimX=nova_cordenada_fim;
```

```
nova_cordenada2_fim = fimY;
```

```
if (nova_cordenada2_fim<1)  
    nova_cordenada2_fim=1;  
end
```

```
if (nova_cordenada2_fim>limY)  
    nova_cordenada2_fim=limY;  
end
```

```
fimY = nova_cordenada2_fim
```

```
%função que determina o centro das regiões
```

```
function centros = determina_centros(imagem_pele_final,etiquetas)
```

```
%centro para cada regioao
```

```
c = regionprops(etiquetas,'Centroid');
```

```
centros=cat(1,c.Centroid);
```

```
%função que determina as dimensões das regiões
```

```
function [coordX,coordY,largura,comprimento] =  
determina_dim(imagem_pele_segmentada)
```



```

%Vai obter a largura e o comprimento da região
%Obtém também a coordenada X e Y do vértice
%do rectângulo inferior mais à esquerda
dim=regionprops(bwlabel(imagem_pele_segmentada),'BoundingBox');
boxdim=cat(1,dim.BoundingBox);
coordX=boxdim(1);
coordY=boxdim(2); %Aqui obtenho as coordenadas para mais
%tarde poder desenhar o rectangulo sobre as faces
largura=boxdim(3);
comprimento=boxdim(4);

%função que efectua as rotações que o template tem que fazer
%para estar com a mesma orientação que a região a classificar

function imagem_media_rotacao = determina_rotacao(imagem_media,angulos,i)

%Se necessário faz a rotação do modelo
if(angulos(i)>0)
imagem_media_rotacao=imrotate(imagem_media,angulos(i)-90);
else
imagem_media_rotacao=imrotate(imagem_media,90+angulos(i));
end
%função que retira a parte do pescoço à imagem

function [comprimento,largura,relacao] =
retira_pescoço(comprimento,largura,dim_X,dim_Y)
comprimento
pause()
% Pre-processamento da imagem para retirar o pescoço
if (dim_X > dim_Y)

    relacao=dim_X/dim_Y;
    if (relacao > 1.25 & relacao<2)

        largura = largura - 0.20*largura;

```

```

end

else

relacao=dim_Y/dim_X;

if (relacao > 1.25 & relacao<2)

    comprimento = comprimento - 0.20*comprimento
    pause ()

end

end
end

```

%Função que efectua o método do template

```

function [imagem_pele_template,dados] =
metodo_template2(imagem0,imagem_pele_final)

%Através da comparação da imagem media de uma face com as possíveis
regiões
%que podem ser uma face obtêm-se o resultado final

%variável onde vão ser armazenadas as coordenadas
%das regiões que contém face
dados=[];

imagem_template=imread('frontal.gif');

[etiquetas,num] = bwlabel(imagem_pele_final,8);

%Determina os ângulos de cada região
angulos=determina_angulos(imagem_pele_final,etiquetas)

%Determina os centros de cada região
centros=determina_centros(imagem_pele_final,etiquetas)

[linha,coluna]=size(imagem_pele_final);

imagem_pele_template=zeros(linha,coluna);%No final esta variável terá a %
%imagem que corresponda apenas a
%faces
imagem_pele_face=zeros(linha,coluna);

```

```

%Adaptar a imagem a compara com as dimensões das possíveis imagens a
%classificar

for i=1:num,

[x,y] = find(etiquetas == i); % coordenadas para as regiões

% obtém a imagem apenas da região
imagem_pele_segmentada = bwselect(imagem_pele_final,y,x,4);

% imshow(imagem_pele_segmentada)
% pause()

%Determina as dimensões e as coordenadas da região a classificar
[coordX,coordY,largura,comprimento]=determina_dim(imagem_pele_segmentada)
;

%Determina a área da região a classificar
area= largura*comprimento;

%obtém o centro da região
coordenada1_centro=centros(i,1);
coordenada2_centro=centros(i,2);

%redimensiona a imagem a comparar com a região a detectar
imagem_media=imresize(imagem_template,[comprimento largura],'nearest');
% imshow(imagem_media)
% pause()

imagem_media_rotacao=determina_rotacao(imagem_media,angulos,i);

%Obtém o centro da imagem media que serve de comparação
imagem_media_final=im2bw(imagem_media_rotacao,0);

%dimensoes da imagem depois dos ajustes
[linha3,coluna3]=size(imagem_media_final);
%ajustar a imagem a comparar
imagem_comparacao=imresize(imagem_media_rotacao,[linha3
coluna3],'nearest');

template_corr = zeros(size(imagem_pele_segmentada));

%Coordenada onde a imagem que serve de comparação vai estar na imagem

inicioX = coordX;
inicioX = round(inicioX);
inicioY = coordY;
inicioY = round(inicioY);
fimX = inicioX + coluna3-1
fimX = abs(round(fimX));

```

```

fimY = inicioY + linha3-1
fimY = abs(round(fimY));

[limY,limX]=size(template_corr);

dim_X=fimX-inicioX
dim_Y=fimY-inicioY

[comprimento,largura,relacao] =
retira_pescoco(comprimento,largura,dim_X,dim_Y);

%Este passo verifica se as coordenadas estão dentro dos limites
%da imagem e caso não estejam atribui-lhe o valor limite mais
%próximo
[inicioX,inicioY,fimX,fimY] =
limites(inicioX,inicioY,fimX,fimY,limX,limY);

size(template_corr);

%Vai obter uma nova imagem com o mesmo tamanho da original mas a imagem
%que serve de comparação nela
for i=inicioY:fimY,
    for j=inicioX:fimX,

        template_corr(i,j) = imagem_comparacao(i-inicioY+1,j-inicioX+1);

    end;
end;

% imshow(imagem_comparacao)
%
% pause()
% imshow(template_corr)
%
% pause()
%
% obter a correlação entre o a imagem a comprar e a região a comparar
correlacao =corr2(template_corr,imagem_pele_segmentada);
%se a correlação é maior que o limiar, a região a classificar é mesmo
%face
if (correlacao>=0.6)
    if (relacao<2)
        if (area>780)

imagem_pele_template=imagem_pele_template+imagem_pele_segmentada;

                dados=[dados coordX coordY largura comprimento];
            end
        end
    end

end

```

```
end
```

```
%Função que efectua a localização final das faces
```

```
%Coloca os rectângulos nas faces
```

```
function imagem0 = desenha_rectangulos(imagem_pele_template,dados,imagem0)
```

```
[c,d]=bwlabel(imagem_pele_template,8);
```

```
dados;
```

```
i=1;
```

```
    set(0,'DefaultRectangleEdgeColor','red')
```

```
imshow(imagem0)
```

```
    if (d>0)
```

```
        hold on
```

```
        for j=1:d
```

```
            imagem_final=rectangle('Position', [dados(i) dados(i+1) dados(i+2) dados(i+3)])
```

```
            i=i+4;
```

```
        end
```

```
    hold off
```

```
    pause()
```

```
end
```