

UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Instituto de Geociências e Ciências Exatas – IGCE
Curso de Bacharelado em Ciências da Computação

CESAR SHUJI FUJIKAWA

Reconhecimento Facial utilizando Descritores de Textura e Aprendizado Não Supervisionado

Orientador: Daniel Carlos Guimarães Pedronette

Rio Claro - SP

2016

Reconhecimento Facial utilizando Descritores de Textura e Aprendizado Não Supervisionado

Relatório de Trabalho de Conclusão de Curso na modalidade Trabalho de Graduação realizado no período integral, apresentado no 2º semestre de 2016 à disciplina ES/TG do Curso de Bacharelado em Ciências da Computação do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de Rio Claro, segundo a Portaria IGCE/DTA no 043/2012.

Aluno: Cesar Shuji Fujikawa

Orientador: Daniel Carlos Guimarães Pedronette

Rio Claro - SP

2016

Resumo

O ser humano é capaz de reconhecer com grande facilidade rostos humanos, mesmo em adversidades como o envelhecimento, má iluminação do local, maquiagem, expressões faciais dentre outros, distinguindo e associando-o com determinada pessoa. Apesar de ser uma tarefa trivial para o nosso cérebro ela é muito complexa, existindo assim uma grande dificuldade para se criar uma abordagem algorítmica e automatizada para isso. Um Sistema de Recuperação de Imagens baseado no Conteúdo (CBIR) tem o objetivo de retornar as imagens mais similares a partir de uma dada imagem de consulta. Os sistemas de reconhecimento facial que podem ser modelados como um sistema CBIR em que, dada uma imagem de rosto humano, retorna uma lista ordenada de imagens com os rostos mais similares dentro de uma determinada base de dados. Neste trabalho utilizamos o descritor de textura Local Binary Patterns para extrair as características das imagens e em seguida utilizamos uma métrica de distância, a euclidiana, para se obter uma lista ordenada de acordo com a similiaridade. Por fim aplicamos uma abordagem não supervisionada de pós processamento baseada em re-ranqueamento (RL-Sim*) que consiste em iterativamente calcular uma nova pontuação baseado em informações contextuais obtidas pela análise de similaridade entre listas de resultados. Uma avaliação experimental foi feita comparando os resultados obtidos antes e depois do algoritmo RL-Sim* e foi observado uma melhora significativa.

Palavras-chave: Reconhecimento Facial; Local Binary Patterns; RL-Sim*; Pós-Processamento; Sistemas CBIR.

Abstract

Humans can easily recognize human faces, even in adversities such as aging, poor lighting of the place, makeup, facial expressions among others, distinguishing and associating it with a certain person. Although it is a trivial task for our brain it is very complex, so there is great difficulty in creating an algorithmic and automated approach to do this. A Content-Based Image Retrieval System (CBIR) aims to return the most similar images from a given query image, facial recognition systems can be considered as a CBIR system, have the ability to given a human face image, return an ordered list of images with the most similar faces within a given database, this is possible thanks to image descriptors and distance metrics. In this work we use an image descriptor, the Local Binary Patterns, which is a texture descriptor, to extract the characteristics of the images and then we use a distance metric, the Euclidean, to obtain an ordered list according to the similiarity, by finally, we apply an unsupervised post-processing re-ranking approach (RL-Sim*), which consists of iteratively calculating a new score based on contextual information obtained by calculating distance metrics. An experimental evaluation was made comparing the results obtained before and after the algorithm RL-Sim* and a significant improvement was observed.

Keywords: Face Recognition; Local Binary Patterns; RL-Sim*; Post Processing; CBIR Systems.

Sumário

1	INTRODUÇÃO	6
2	TRABALHOS RELACIONADOS	9
2.1	Correlação	9
2.2	Eigenfaces	9
2.3	Fisherface	12
2.4	Classificadores kNN	14
2.5	Classificadores por distância	15
2.5.1	Métrica de Distância Euclidiana	15
2.5.2	Métrica de Distância Euclidiana Quadrada	16
2.5.3	Distância Manhattan	16
2.5.4	Distância Chebyshev	16
3	RECUPERAÇÃO DE IMAGENS E DESCRITORES DE TEXTURA	17
3.1	Recuperação de Imagens	17
3.2	Descritores de Imagens	17
3.3	Descritores de Textura	18
3.4	Local Binary Patterns	19
4	APRENDIZADO NÃO-SUPERVISIONADO	21
4.1	RL-Sim	21
4.1.1	Descrição Formal do algoritmo RL-Sim	21
4.2	RL-Sim*	22
5	AVALIAÇÃO EXPERIMENTAL	25
5.1	Datasets	25
5.1.1	The ATT Face Database (The ORL Database of Faces)	25
5.1.2	Extended Yale Facedatabase B	26
5.1.3	The Sheffield Face Database (The UMIST Face Database)	26
5.2	Métricas de Eficácia	26
5.2.1	Precisão	27
5.2.2	MAP (Mean Average Precision)	27
5.2.3	Recall (Revocação)	28
5.3	Resultados	28
5.3.1	Métrica P@20; MAP	28
5.3.2	Definição dos parâmetros	28

5.3.3	Curvas Precision x Recall	31
6	CONCLUSÃO	37
	REFERÊNCIAS	38

1 Introdução

Nos dias atuais, com a crescente demanda de informações, como por exemplo, fotos, imagens e vídeos, houve também uma grande necessidade de reconhecimento de indivíduos em locais com alta concentração de pessoas, como shoppings, aeroportos, estações, entre outros. Diversas aplicações que necessitam de uma boa eficiência em reconhecimento, como por exemplo na área de segurança, onde é necessário identificar algum criminoso ou um visitante não autorizado em áreas restritas (CARNEIRO, 2012), utilizando biometria facial. Essas tarefas são de extrema importância embora não seja viável utilizar o humano para analisar toda essa quantidade de dados. A relevância em pesquisas para reconhecimento facial vem crescendo cada vez mais na época em que vivemos.

Nós humanos temos uma grande capacidade em reconhecer rostos, conseguimos identificar cada detalhe e facilmente associamos o rosto com uma pessoa, mesmo em condições desfavoráveis como o uso de acessórios, a iluminação do local, envelhecimento, barba, expressões faciais, maquiagem, entre outros (CARNEIRO, 2012). Porém a maneira como o nosso cérebro interpreta e processa as informações para esse reconhecimento não é trivial, assim sendo uma tarefa complexa para criarmos abordagens algorítmicas e automatizadas e implementá-las em máquinas.

Em contrapartida, abordagens automatizadas têm sido aplicadas para auxiliar na organização da grande quantidade de conteúdo visual disponível atualmente. Por exemplo, um sistema CBIR (Content-based Image Retrieval) (FUJIKAWA; PEDRONETTE, 2014) consiste em dada uma imagem de consulta, retornar as imagens mais similares dentro de uma coleção de imagens baseadas em seu conteúdo visual. Os sistemas de reconhecimento facial podem ser modelados como um sistema CBIR, ou seja é possível dada uma imagem de rosto humano, retornar uma lista ordenada de imagens com os rostos mais similares dentro de uma base de dados. Em geral, essas tarefas são realizadas utilizando abordagens de extração de características de imagens e métricas de distância que compara pares de imagens.

Entretanto, mesmo após anos de pesquisa, ainda é uma tarefa árdua fazer o reconhecimento de rostos de forma automatizada de maneira eficaz. Sendo assim, métodos de aprendizado não-supervisionados tem sido propostos com o objetivo de aumentar a eficácia de tarefas de recuperação de imagens realizadas pelos sistemas CBIR. Esses métodos exploram informações contextuais a partir de relacionamentos entre as imagens calculando uma nova métrica de distância mais precisa.

O objetivo deste trabalho consiste na utilização de algoritmos de reconhecimento de faces em Sistemas CBIR e a aplicação de métodos de aprendizado não supervisionado

para aumentar a eficácia dos resultados.

O procedimento utilizado neste trabalho foi ilustrado na Figura 1, inicialmente aplicamos o algoritmo Local Binary Patterns (AHONEN; HADID; PIETIKAINEN, 2006), que é um dos mais conhecidos descritores de textura, em três diferentes coleções de imagens em seguida foi aplicado o algoritmo RL-Sim* (OKADA; PEDRONETTE; TORRES, 2015), um método não supervisionado, cujo intuito é melhorar a eficácia de recuperação de imagens desses sistemas, baseando-se em um processo reclassificação iterativa (*re-ranking*) (PEDRONETTE; TORRES, 2012). Para isso, são analisadas informações contextuais codificadas nas listas de resultados (*ranked lists*), e não somente entre um par de imagens.

Após a aplicação do método foi realizada uma avaliação experimental que comparou os resultados obtidos pelo algoritmo LBP e em seguida com a aplicação do algoritmo RL-Sim*, possibilitando avaliar os ganhos ou perdas nos resultados de maneira objetiva.

Este trabalho está dividido em: na Seção 2 é discutido alguns trabalhos relacionados ao reconhecimento de faces, tanto abordagens já existentes como tipos de classificadores por distância, na Seção 3 apresenta toda introdução e todo o funcionamento do algoritmo Local Binary Patterns, na seção 4 apresenta o algoritmo RL-Sim*, abordagem de aprendizado não supervisionado utilizado neste trabalho. Na seção 5 é discutido as coleções de imagens utilizadas, as métricas de eficácias e por fim os resultados obtidos, por fim a seção 6 apresenta a conclusão obtida após todo o trabalho e a análise dos resultados obtidos.

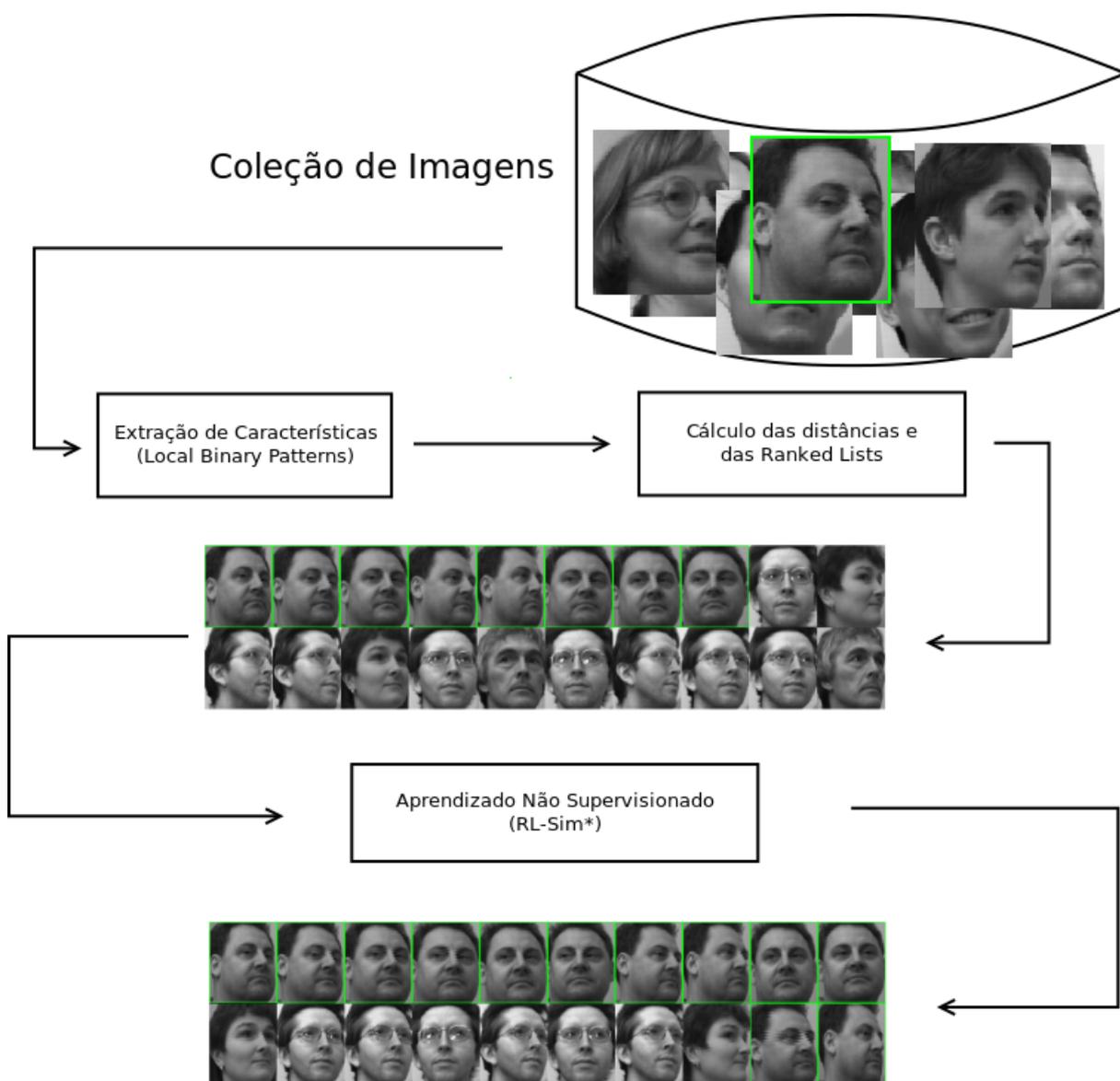


Figura 1 – Reconhecimento facial baseado em descritores de textura e aprendizado não supervisionado.

2 Trabalhos Relacionados

Nesta seção serão apresentadas algumas das abordagens mais comumente usadas em reconhecimento de faces. Primeiramente serão apresentadas abordagens de extração de vetores como o Eigenfaces (TURK; PENTLAND, 1991) e o Fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997), em seguida serão discutidos classificadores baseados em vizinhança.

2.1 Correlação

É a abordagem de classificação mais simples, similar à classificação por vizinhança no espaço da imagem (BELHUMEUR; HESPANHA; KRIEGMAN, 1997). As imagens são classificadas pela distância mais próxima entre os pontos nas imagens, considerando que A e B são duas imagens distintas, a partir da comparação de um ponto A_{ij} com um ponto B_{ij} , é possível aplicar uma métrica de distância para verificar a similaridade entre elas. Pela simplicidade, esta abordagem possui muitas desvantagens. A primeira é que se as imagens de uma base de dados apresentarem uma alta variação de luz, os pontos correspondentes terão distâncias com altas variações, logo esses pontos não poderão ser bem agrupados. Para esse método trabalhar com segurança, é preciso de uma base de dados que tenha uma grande permanência da mesma condição de luz para todas as imagens. A segunda é que o método da correlação possui um alto custo computacional, pois nenhum tipo de redução é feito, fazendo com que haja um custo muito grande dependendo do tamanho da imagem. E por fim, a terceira desvantagem deste método consiste na necessidade de uma grande amostra de imagens para cada indivíduo para o seu adequado funcionamento.

2.2 Eigenfaces

Como os métodos de correlação são métodos computacionalmente caros e requerem uma grande quantidade de imagens no *dataset*, então é desejável que se reduza a dimensão dos dados. Uma técnica que está sendo comumente usada para a redução de dimensionalidade na área de visão computacional, particularmente em reconhecimento de faces é o PCA (*Principal Component Analysis*) (BARAHATE; SATURWAR, 2010). Técnicas PCA também conhecidas como métodos Karghunen-Loeve, reduzem a dimensionalidade dos dados através de uma análise estatística que se baseia na redundância e na variância existentes nos dados, ao aplicar o PCA em reconhecimento de faces, é chamado de Eigenface. O Eigenface é baseado na projeção linear da imagem em um espaço de baixa dimensionalidade, podemos ver um exemplo na Figura 3. Um problema em conjuntos de

dados multidimensionais é representar os dados em um sub-espço de baixa dimensão, para resolvê-lo é preciso representar estes mesmos dados de modo em que a representação em uma dimensão mais baixa corresponda o máximo possível entre os pontos no espaço dimensional original. A técnica PCA é uma das mais famosas para resolver esse problema, ela tem como objetivo encontrar um sub-espço contendo bases ortonormais, os principais vetores que definem a distribuição dos dados que otimizarão a correlação entre os dados. A Figura 2 ilustra como é feito a extração destes vetores:

Exemplo PCA: Eigenfaces

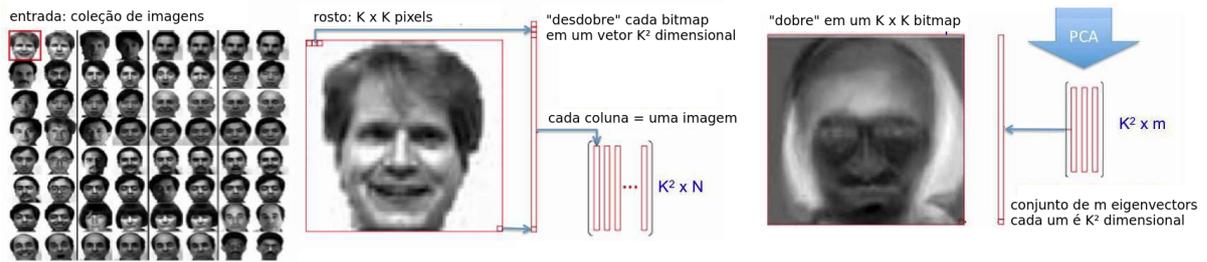


Figura 2 – Principal Component Analysis.

(EDINBURGH, 2014)

Um ponto inconveniente de se utilizar o Eigenfaces (TURK; PENTLAND, 1991) é que, quando o PCA (BARAHATE; SATURWAR, 2010) reduz a dimensionalidade ela faz com que as projeções maximizem as dispersões entre as classes, mas não só a dispersão entre classes (que é a útil para a classificação), mas também a dispersão dentro da própria classe da imagem, que é um efeito indesejável para a classificação. Assim caso a base de dados contiver imagens de rostos variando sob a iluminação, a projeção terá muitas características alteradas graças a iluminação, assim diminuindo a eficácia no agrupamento dos dados.

Para explicar formalmente o Eigenface consideraremos um conjunto de N imagens (x_1, x_2, \dots, x_N) , que formam o grupo de vetores que se associam com os pixels uma imagem, primeiramente é calculado uma média das imagens utilizando a equação:

$$Média = 1/N \cdot \sum_{n=1}^N x_n \quad (2.1)$$

Após isso, calcula-se os desvios de cada imagem em relação a média calculada

$$\rho_i = x_i - Média \quad (2.2)$$

Esse conjunto de vetores que está definido por ρ , está normalizado em relação à média e está pronto para ser usado no cálculo de autovalores α_k , e autovetores β_k , que forma o conjunto de N vetores ortonormais, seguindo a equação:

$$\alpha_k = 1/N \cdot \sum_{n=1}^N (\beta_k^T \cdot \rho_n)^2 \quad (2.3)$$

o α_k , é sujeito a seguinte equação:

$$\beta_l^T \cdot \beta_k = \begin{cases} 1 & \text{se } l = k; \\ 0 & \text{se } l \neq k. \end{cases} \quad (2.4)$$

Os vetores β_k e escalares α_k são autovetores e autovalores respectivamente da matriz de covariância da seguinte equação:

$$C = 1/N \cdot \sum_{n=1}^N \rho_n \rho_n^T = A \cdot A^T \quad (2.5)$$

sendo a matriz A o conjunto de vetores normalizado em relação à média $A = (\rho_1, \rho_2, \dots, \rho_N)$.

A matriz C é de $M^2 \times M^2$ e M^2 determina os autovetores e autovalores, que é inviável para o tamanho das imagens. Assim é necessário reduzir a dimensionalidade do M^2 , considerando V_i os autovetores de A^T com a seguinte equação:

$$A^T \cdot A \cdot V_i = \alpha_i \cdot V_i \quad (2.6)$$

multiplicando os dois lados por A resulta em:

$$A \cdot A^T \cdot A \cdot V_i = A \cdot \alpha_i \cdot V_i \quad (2.7)$$

assim os autovetores de $C = A \cdot A^T$ serão igual a $A \cdot V_i$.

Desta forma é gerado a matriz de dimensão $M \times M$, conseqüentemente reduzindo o número de computações de M^2 para N número de imagens, fazendo com que o conjunto de treinamento fique consideravelmente menor.



Figura 3 – Representação de uma imagem após o Eigenfaces.

(TRAN, 2008)

2.3 Fisherface

Historicamente o Fisherface (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) foi empregado em sistemas de reconhecimento de fala, posteriormente começou a ser aplicado em sistemas de reconhecimento de face e pode ser visto na Figura 4. O Fisherface é uma técnica que visa maximizar a dispersão entre as classes, e sua maior diferença em relação ao Eigenfaces (TURK; PENTLAND, 1991), é que ele também reduz a dispersão dentro da própria classe, características que não ocorria no Eigenfaces, uma vez que também se aumentava a dispersão dentro da própria classe, podendo ocasionar em um resultado indesejável. Basicamente o algoritmo Fisherface tenta encontrar a melhor projeção em retas que maximizem a separabilidade dos escalares.

Formalmente considere duas classes distintas em um ambiente 2D, suponha que um conjunto de N imagens (x_1, x_2, \dots, x_N), que são distribuídas nessas classes, a classe 1 e 2 denominaremos respectivamente w_1 e w_2 , com M_1 e M_2 imagens. Para encontrar o melhor vetor para projeção, o vetor médio de cada classe no espaço original e no espaço Fisherfaces é obtido pelas equações:

$$\alpha_i = 1/N_i \cdot \sum_{x \in w_i} x \quad (2.8)$$

$$\hat{\alpha}_i = 1/N_i \cdot \sum_{y \in w_i} y = 1/N_i \cdot \sum_{x \in w_i} W^T \cdot X = W^T \cdot \alpha_i \quad (2.9)$$

O Fisherface (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) tenta maximizar a dispersão que é resultante da diferença entre as média normalizadas inerente a cada classe, a dispersão que é equivalente a variância é representado por:

$$\hat{S}_i^2 = \sum_{y \in w_i} (y - \hat{\alpha}_i)^2 \quad (2.10)$$

O vetor de projeção das amostras é representado por W , que possui as características de uma mesma classe e faz com que permaneçam próximas e também deixando as médias entre os outros conjuntos mais distantes, o Fisherface é considerado como uma função linear $W^T.X$ que maximiza a função representado pela equação abaixo:

$$J(W) = \sum_{y \in w_i} (y - \hat{\alpha}_i)^2 \quad (2.11)$$

O objetivo do vetor W é encontrar uma melhor projeção, assim torna-se necessário expressar $J(W)$ como uma função de W como na seguinte equação:

$$S_I = \sum_{x \in w_i} (x - \alpha_i).(X - \alpha_i)^T \quad (2.12)$$

Como o Fisherfaces é baseado em selecionar os vetores W que possuem as características semelhantes de uma mesma classe e formar um espaço vetorial que a dispersão entre as classes e dentro das classes sejam melhoradas, a matriz de dispersão S_W é dada por $S_W = S_1 + S_2$ e a matriz de dispersão entre classes e dentro das classes é dada respectivamente pelas equações:

$$S_B = \sum_{i=1}^c N_i.(\alpha_i - \alpha).(\alpha_i - \alpha)^T \quad (2.13)$$

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \alpha_i).(X_k - \alpha_i)^T \quad (2.14)$$

α é a média de todo o conjunto, α_i é a média das imagens de classe X_i , e o número de amostras na classe X_i é dado por N_i . O vetor de projeção W_{otimo} é escolhido como a matriz com colunas ortonormais, caso S_w não seja único. O W_{otimo} maximiza a razão entre o determinante da matriz de espalhamento entre classes com a determinante da matriz de espalhamento dentro das classes, isso se dá com a seguinte equação:

$$W_{otimo} = argmax(W^T.S_B.W)/(W^T.S_W.W) = [w_1, w_2, \dots, w_m] \quad (2.15)$$

sendo $\{W_i | i = 1, 2, \dots, m\}$ o conjunto generalizado de autovetores S_B e S_W que correspondem aos m maiores autovalores.

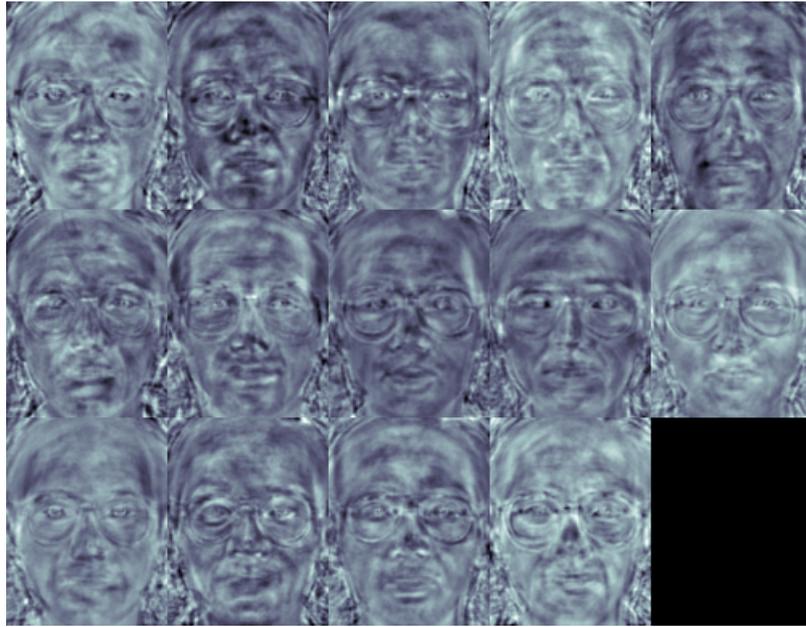


Figura 4 – Representação de imagens com o Fisherface.

(ZHAO et al., 2003)

2.4 Classificadores kNN

Os classificadores kNN (*k-Nearest Neighbors*) (SOHAIL; BHATTACHARYA, 2007) são baseados em buscar os k elementos mais similares ou de menor distância dentro de um conjunto de treinamento, a classe que tiver a maior frequência dentro do modelo de treinamento será atribuído ao elemento de consulta. O único parâmetro livre é o k que define o número de k -vizinhos que serão consultados, ele é controlado pelo usuário e tem o objetivo de melhorar a eficácia da classificação. Podemos visualizar melhor o conceito com a Figura 5, onde elementos da classe em preto são mais comuns entre os vizinhos mais próximos e são selecionados para determinar a classificação.

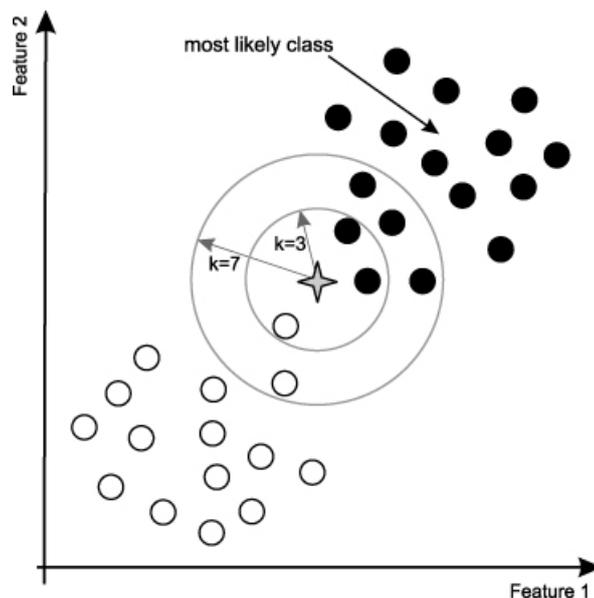


Figura 5 – Imagem ilustrando um Classificador kNN.

(PARDESHI, 2013)

Podemos perceber que o elemento dado será classificado de acordo com k , verificando qual a maior frequência da classe dos vizinhos mais semelhantes. Veremos na próxima seção um pré requisito para estabelecermos a classificação kNN, os classificadores por distância.

2.5 Classificadores por distância

Muitas das abordagens de reconhecimento de face da última década tomam como base a medida de distância (RADY, 2011). As imagens são projetadas sob um espaço de características de baixa dimensionalidade. As distâncias entre as representações do espaço de características são utilizados para tomar as decisões de reconhecimento. As medidas de distância são os últimos componentes para se fazer o reconhecimento de faces entre os vetores de duas imagens provê informações sobre a similaridade entre elas. Nas próximas sub-seções serão discutidas algumas métricas de distância.

2.5.1 Métrica de Distância Euclidiana

Uma das abordagens mais recorrentes para se calcular a distância entre duas imagens é a distância euclidiana (RADY, 2011). Quando as pessoas se referem a simplesmente "distância" ou "distância simples" na realidade estão se referindo a Distância Euclidiana, que é definida basicamente como a raiz quadrada do somatório de 0 até N , o número total.

$$D.E(A, B) = \sqrt{\sum_{i=1}^N (A_i - B_i)^2} \quad (2.16)$$

Ela pode ser provada por iterações repetidas do Teorema de Pitágoras, ao classificar a distância entre uma dada imagem de consulta e uma imagem contida dentro de um banco de dados, quanto menor o resultado da distância calculada, maior será a similaridade entre as duas imagens, assim a imagem que receber o resultado com o menor número provavelmente será a imagem mais similar.

2.5.2 Métrica de Distância Euclidiana Quadrada

É extremamente similar a Métrica de Distância Euclidiana normal, mas ela possui uma variação, a raiz quadrada do somatório entre a diferença dos quadrados dos elementos dos vetores que representam as imagens A e B permanece. Segue a equação:

$$D.E.Q(A, B) = \sum_{i=1}^N (A_i - B_i)^2 \quad (2.17)$$

2.5.3 Distância Manhattan

Distância Manhattan, *City-Block Distance* ou distância retilínea (RADY, 2011), representa a distância entre pontos como se fosse em formatos horizontais e verticais, uma analogia seria ruas ou estradas que carros passam, em contrapartida da distância euclidiana que seria apenas uma reta, ela examina a diferença entre um par de imagens com a seguinte equação:

$$C.B.D(A, B) = \sum_{i=1}^N (|A_i - B_i|) \quad (2.18)$$

2.5.4 Distância Chebyshev

Distância Chebyshev ou distância máxima (RADY, 2011), é uma métrica definida em um espaço vetorial que a distância é escolhida a partir da maior diferença entre dois vetores ou pontos ao longo de todas as coordenadas. A Distância Chebyshev entre duas imagens A e B é:

$$D.C(A, B) = \max_i (|A_i - B_i|) \quad (2.19)$$

3 Recuperação de Imagens e Descritores de Textura

Nesta seção será apresentada uma introdução a respeito de recuperação de imagens e também de descritores de imagens, será abordado especificamente descritores de textura e o funcionamento e descrição do algoritmo Local Binary Patterns.

3.1 Recuperação de Imagens

A recuperação de informações em um contexto geral é uma área da computação que é responsável por recuperar informações e dados úteis de acordo com um contexto, de forma automatizada dentro de uma coleção de dados. A classificação de informações é responsável por separar a informação já recuperada e organizá-la em diferentes contextos, categorias ou grupos, assim possibilitando a manipulação dos dados. Aplicando a recuperação de dados e a classificação no contexto de recuperação de imagens, encontramos um cenário, dada uma imagem de consulta organizar uma base de imagens de forma a retornar uma lista ordenada de imagens (*ranked lists*), conforme sua similaridade no conteúdo visual (FUJIKAWA; PEDRONETTE, 2014). Essa similaridade é avaliada de acordo com diferentes propriedades visuais, e é calculada por um descritor de imagens.

3.2 Descritores de Imagens

Uma típica arquitetura CBIR, representado na Figura 6, precisa de um descritor de imagem, que é caracterizado por: (i) um algoritmo de extração de características, que codifica as características das imagens em vetores e (ii) uma métrica de similaridade que compare duas imagens. Uma métrica de similaridade nos fornece um grau de similaridade dado um par de imagens, representadas por seus vetores de características, também definido como uma função inversa de distância (e.g. Distância Euclidiana). (FALCAO; TORRES, 2006)

A extração de características é a base de toda recuperação de imagens baseados em conteúdo visual, ela é subjetiva, uma vez que existem diversos modos de representar uma mesma particularidade de uma determinada imagem, existem dois tipos de escopos para a classificação de características, o domínio geral, domínio específico. (SILVA, 2016) O domínio geral avalia característica de cor textura e forma, enquanto o domínio específico envolve conhecimento e características específicas particulares ao problema em questão, como padrões de reconhecimento de faces, classificação de impressões digitais, placas de

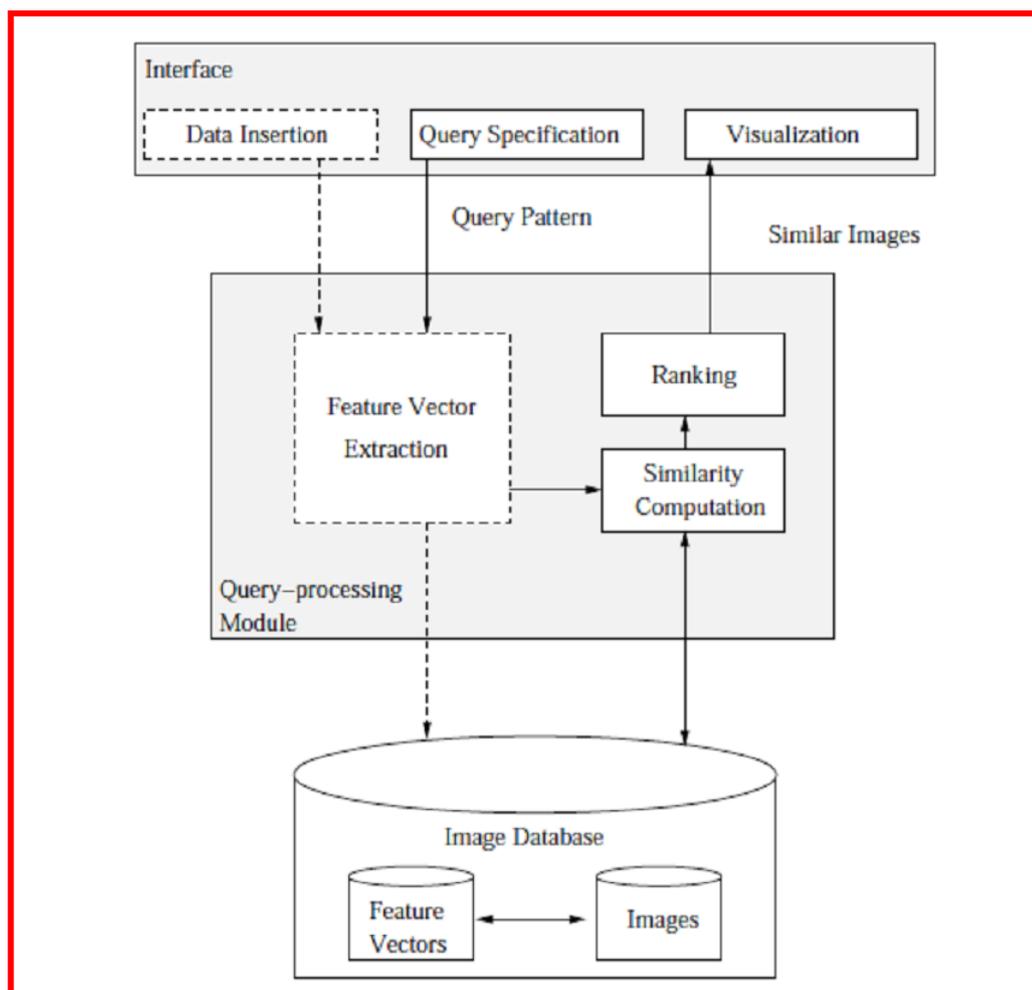


Figura 6 – Representação de uma arquitetura CBIR.
(FUJIKAWA; PEDRONETTE, 2014)

veículos dentre outros. Ainda existem dois tipos de descritores de imagens, os locais e os globais, os descritores locais são baseados e calculados em regiões específicas ao redor dos pontos de interesses, normalmente possuem uma coordenada 2D na imagem que será o ponto central, uma orientação, uma escala e uma região característica ao redor da coordenada. Os descritores de imagens globais são a abordagem mais tradicional de classificação de imagens, estes consideram a imagem como um todo, não sendo analisado região por região isoladamente, descrevendo as imagens baseadas nas informações contextuais como cor textura e forma assim como na local.

3.3 Descritores de Textura

A textura está presente em todas os materiais existentes (SILVA, 2016) e é facilmente reconhecida em imagens, pode ser visualizada em elementos como areia, folhas, nuvens, tijolos, etc. Ela contém informações importantes sobre a superfície de um material e frequentemente vem sendo usada em algoritmos de processamento de imagens e visão

computacional dando excelentes resultados em diversas aplicações desde diagnóstico de imagens médicas até o reconhecimento facial de humanos.

A textura é um elemento difícil de ser analisado se considerarmos apenas o valor de um pixel, a análise ocorre principalmente considerando a variação do valor dos pixels da sua própria vizinhança. Isso torna possível nomear alguns atributos como rugosidade, contraste, direcionalidade, regularidade e probabilidade de linha, sendo os três primeiros os mais relevantes para a análise. (PENATTI; VALLE; TORRES, 2012)

3.4 Local Binary Patterns

O Local Binary Patterns (AHONEN; HADID; PIETIKAINEN, 2006) é um descritor local de textura, atualmente é um dos mais utilizados, recentemente tem sido muito utilizado em aplicações de reconhecimento facial, ou qualquer aplicação que necessite de uma análise de textura. A idéia básica é considerar um parâmetro N para utilizar uma vizinhança $N \times N$ pixels, considerando-se $N = 3$ assim como utilizamos neste trabalho, existem 8 vizinhos e um ponto central, $V = \{V_0, V_1, V_2, \dots, V_8\}$, o ponto central é representado por V_0 , para cada V_i é representado uma cor em escala de cinza que pode receber um score de 0 a 256. O LBP é representado por $LBP = \{E_1, E_2, \dots, E_8\}$, sendo E_i sendo obtido a partir da comparação entre o ponto central V_0 e seus vizinhos V_i . A comparação consiste em dois possíveis resultados, se o pixel central V_0 possuir um valor maior ou igual que o pixel V_i , então E_i será igual a 1, se V_0 possuir um valor menor que V_i , então E_i terá valor igual a 0. Após calculada a nova matriz de pixels, esta será multiplicada ponto a ponto por uma matriz contendo pesos, essa matriz de pesos é obtida pela equação:

$$M_{peso} = \sum_{i=1}^8 \left(E_i * 2^{i-1} \right) \quad (3.1)$$

e por fim soma-se todos os resultados que será utilizado para formar um histograma, que é considerado um vetor de características que define o padrão analisado, na Figura 7 podemos ver o passo a passo, na Figura 8 podemos verificar o antes o depois da aplicação do algoritmo.

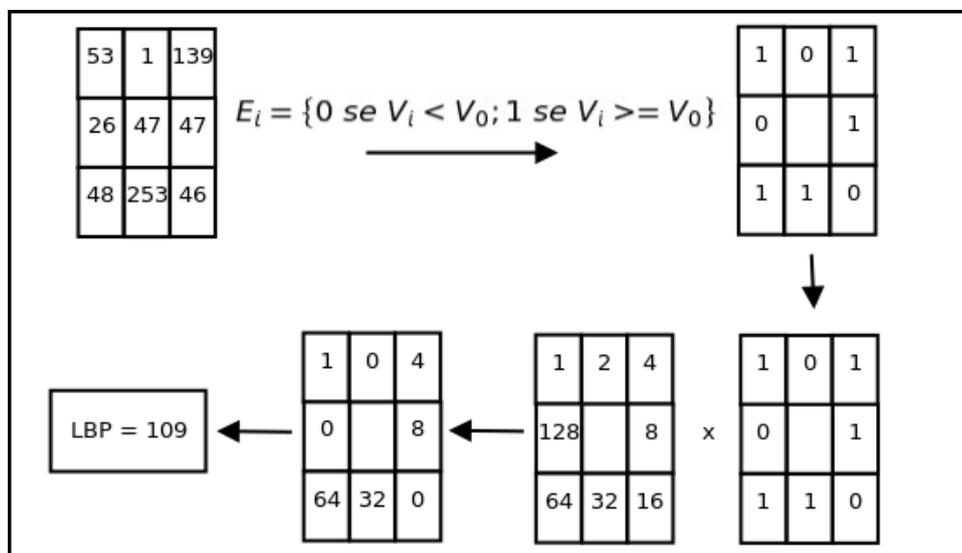


Figura 7 – Exemplo de cálculo do Local Binary Patterns.



Figura 8 – Exemplo de antes e depois da aplicação do LBP

4 Aprendizado Não-Supervisionado

As seções seguintes apresentam uma discussão sobre métodos aprendizado não-supervisionado. As seções 4.1 e 4.2 discutem os algoritmos RL-Sim e RL-Sim*, utilizado neste trabalho.

4.1 RL-Sim

O algoritmo RL-Sim (PEDRONETTE et al., 2013) é um método de aprendizagem não supervisionado, cujo objetivo consiste em explorar o conjunto inicial de *ranked lists* por meio de um processo de reclassificação iterativa (*re-ranking*) (PEDRONETTE; TORRES, 2012), observando informações contextuais, e com elas gerar um novo conjunto de *ranked lists* mais eficazes que serão utilizados numa próxima iteração. Assim o processo se repetirá gerando resultados cada vez mais eficazes, ao fim das iterações uma nova e definitiva *ranked list* é criada. Uma particularidade da abordagem, é que ao invés de analisar pares de imagens como a maioria das abordagens, são analisadas as *ranked lists* que são ricas em informações contextuais. Quando duas imagens apresentam alto grau de similaridade, na maioria das vezes é provável que seus respectivos *ranked lists* também apresentem certa similaridade. Assim o algoritmo RL-Sim por meio de um cálculo da similaridade entre as *ranked lists*, melhora a eficácia das distâncias somente utilizando a informação do *ranking* da imagem, sem requerer informações de distância.

4.1.1 Descrição Formal do algoritmo RL-Sim

A métrica contextual de distância descrita pelo RL-Sim (PEDRONETTE et al., 2013) tem como base a suposição que as imagens que possuem uma melhor classificação dentro de uma *ranked list*, também possuem uma alta similaridade entre si. Com esta base, é iterativamente calculado um novo score através das informações contextuais fornecidas por métricas de correlação de *rankings*.

Seja $\mathcal{N}(i, k)$ uma vizinhança, o i correspondendo a uma imagem img_i e k correspondendo a quantidade de imagens mais similares da imagem img_i , e que a vizinhança foi determinada de acordo com uma distância ρ gerada por um descritor de imagem, o conjunto $\mathcal{N}(i, k)$ pode ser obtido pelos k vizinhos mais próximos, e a cardinalidade do conjunto é definido por $|\mathcal{N}(i, k)| = k$.

Seja $d(\tau_i, \tau_j, k)$ uma métrica de correlação entre *ranked lists*, considerando τ_i e τ_j , *ranked lists* de i e j , utiliza-se as suas primeiras posições definida pelo conjunto $\mathcal{N}(i)$ e $\mathcal{N}(j)$, que são definida no intervalo entre $[0, 1]$. Pode ser definida uma métrica de distância

$\rho_c(img_i, img_j)$, baseada na comparação dos *ranked lists* τ_i, τ_j como:

$$\rho_c(img_i, img_j) = d(\tau_i, \tau_j, k) \quad (4.1)$$

Baseando-se na conjectura de que a métrica de distância ρ_c representa uma distância mais refinada e eficaz entre as imagens, é possível recalculá-la a distância entre as imagens e também a distância contextual de forma iterativa, de forma que se gere resultados melhores a cada iteração.

Seja (t) a iteração atual do algoritmo e seja $\tau_i^{(t)}$ o *ranked list* da iteração t . Seja $\rho_c^{(0)}$ a distância contextual entre imagens na primeira iteração, a qual é igual a distância definida pelo descritor de imagem, tal que $\rho_c^{(0)}(img_i, img_j) = \rho(img_i, img_j)$ para todas as imagens $img_i, img_j \in \mathcal{C}$. A métrica iterativa é definida como:

$$\rho_c^{(t+1)}(img_i, img_j) = d(\tau_i^{(t)}, \tau_j^{(t)}, k) \quad (4.2)$$

É esperado que a acurácia da distância melhore ao longo das iterações, removendo as imagens não relevantes das primeiras posições das *ranked lists*. Desta forma, o tamanho da vizinhança k pode ser incrementada, sendo possível considerar mais imagens ao longo das iterações. Considerando este aumento, a métrica pode ser redefinida como:

$$\rho_c^{(t+1)}(img_i, img_j) = d(\tau_i^{(t)}, \tau_j^{(t)}, k + t) \quad (4.3)$$

Depois de um determinado número T de iterações, uma nova distância é calculada baseada na métrica de distância contextual:

$$\hat{\rho}(img_i, img_j) = \rho_c^{(T)}(img_i, img_j) \quad (4.4)$$

Por fim, usando a distância gerada $\hat{\rho}$, uma nova matriz de distância pode ser calculada como $\hat{A}_{ij} = \hat{\rho}(img_i, img_j)$. Baseada em \hat{A} , um novo e definitivo conjunto de *ranked lists* $\hat{\mathcal{R}}$ pode ser calculado também.

4.2 RL-Sim*

O RL-Sim (PEDRONETTE et al., 2013) considera as top- k posições calculadas no conjunto $\mathcal{N}(i, k)$ e $\mathcal{N}(j, k)$ assim determinando a nova distância entre as img_i e img_j .

Desta maneira, a redefinição das distâncias é calculada com base nas imagens do início dos *ranked lists* e não mais nos *ranked lists* inteiros.

O motivo dessa abordagem é que, considerando que provavelmente as imagens mais similares estarão no topo de uma *ranked list* e as não similares estarão no fim, são avaliadas

as L primeiras distâncias que vão possuir uma acurácia maior, então provavelmente as distâncias redefinidas com as L primeiras, terão também uma eficácia maior.

Entretanto mesmo resultando numa boa eficácia e um baixo custo computacional, existem limitações nesta abordagem, como o cálculo de correlações entre as posições que não existem interseção dentro dos conjuntos $\mathcal{N}(i, k)$ e $\mathcal{N}(j, k)$, assim essas que não possuem interseção tendem a ter uma pontuação menor do que a merecida, e conseqüentemente a eficácia da distancia é piorada.

Para contornar essa situação o algoritmo RL-Sim* (OKADA; PEDRONETTE; TORRES, 2015), criou modos diferentes de calcular a distância quando não há interseção, de forma que ao receber uma imagem de consulta img_i , seu *ranked list* será dividido em três segmentos que serão analisados e processados de formas diferentes:

(i) Primeiro Segmento (top- L posições, com a interseção): se a vizinhança do conjunto img_i, img_j possuir imagens em comum. então o segmento contém uma imagem img_j nas top- L . Formalmente podemos descrever $(\tau_i(j) < L) \wedge (\mathcal{N}(i, k) \cap \mathcal{N}(j, k) \neq \emptyset)$. Nesse primeiro segmento a nova distância entre a imagem img_i e img_j é calculada através da métrica de correlação definida no intervalo de $[0, 1]$, da mesma forma que era feito no algoritmo RL-Sim (PEDRONETTE et al., 2013) original.

(ii) Segundo Segmento (top- L posições, sem a interseção): se a vizinhança do conjunto img_i, img_j não possuir imagens em comum entre as top- k posições ($\mathcal{N}(i, k) \cap \mathcal{N}(j, k) = \emptyset$), a métrica de correlação não poderá nos fornecer informações sólidas para melhorar a medida de distância. Nesses casos, soma-se 1 a distância, de forma que $A_{ij}^{(t+1)} = A_{ij}^{(t)} + 1$, continuando com sua ordem original.

(iii) Restante das Imagens: nas distâncias das imagens em que obtiveram uma pontuação menor, e se posicionaram abaixo top- L , soma-se 2 a distância, de forma que $A_{ij}^{(t+1)} = A_{ij}^{(t)} + 2$. O objetivo disso é de manter essas imagens no fim dos *ranked lists*, garantindo que elas não se misturem com as imagens com maior probabilidade de estarem correlatas.

O Algoritmo 1 descreve a proposta do Algoritmo RL-Sim* (OKADA; PEDRONETTE; TORRES, 2015).

Algorithm 1 Algoritmo RL-Sim***Require:** Matriz de distância A , Conjunto de *ranked lists* \mathcal{R} e parâmetros k, T, L **Ensure:** Conjunto preprocessado de *ranked lists* $\hat{\mathcal{R}}$

```

1:  $t \leftarrow 0$ 
2:  $k \leftarrow k_s$ 
3:  $\mathcal{R}^{(t)} \leftarrow \mathcal{R}$ 
4:  $A^{(t)} \leftarrow A$ 
5: while  $t < T$  do
6:   for all  $\tau_i \in \mathcal{R}^{(t)}$  do
7:     for all  $\{img_j \in \mathcal{C} \mid \tau_i(j) \leq L\}$  do
8:       if  $\mathcal{N}(i, k) \cap \mathcal{N}(j, k) \neq \emptyset$  then
9:         //Primeiro Segmento -Com Sobreposição
10:         $A^{(t+1)}[i, j] \leftarrow d(\tau_i, \tau_j, k)$ 
11:       else
12:         //Segundo Segmento - Sem Sobreposição
13:         $A_{ij}^{(t+1)} \leftarrow A_{ij}^{(t)} + 1$ 
14:       end if
15:     end for
16:     //Imagens Remanescentes - Depois de top- $L$  Posições
17:     for all  $\{img_j \in \mathcal{C} \mid \tau_i(j) > L\}$  do
18:        $A_{ij}^{(t+1)} \leftarrow A_{ij}^{(t)} + 2$ 
19:     end for
20:   end for
21:    $\mathcal{R}^{(t+1)} \leftarrow \text{sortRankedLists}(A^{(t+1)})$ 
22:    $k \leftarrow k + 1$ 
23:    $t \leftarrow t + 1$ 
24: end while
25:  $\hat{\mathcal{R}} \leftarrow \mathcal{R}^{(T)}$ 

```

5 Avaliação Experimental

Esta seção apresenta os descritores, as coleções de imagens utilizadas e os resultados da avaliação experimental realizada. Utilizamos o Local Binary Patterns (AHONEN; HADID; PIETIKAINEN, 2006) que é um descritor de textura local. Para a avaliação ainda foram consideradas três coleções de imagens diferentes, o The ATT Face Database (The ORL Database of Faces) (CAMBRIDGE, 2002), o Extended Yale B Face Database (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001) e por fim o The Sheffield Face Database (GRAHAM; ALLINSON, 1998) (antigo The UMIST Face Database).

5.1 Datasets

As seções seguintes descrevem as coleções de dados utilizadas na avaliação experimental.

5.1.1 The ATT Face Database (The ORL Database of Faces)

A base de dados ATT Face Database (CAMBRIDGE, 2002), representado na Figura 9, originalmente The ORL Database of Faces, contém um conjunto de imagens coletadas entre abril de 1992 e abril de 1994. Existem 10 imagens para cada um dos 40 indivíduos, as imagens possuem variação de tempo, variação de luz, variação de expressões faciais (olhos abertos/fechados, sorrindo/não sorrindo) e alguns detalhes como óculos e sem óculos. Todas as imagens foram tiradas com um fundo escuro homogêneo com os indivíduos em posição frontal. No total as são 400 imagens em preto e branco (256 escalas de cinza), o tamanho é 92×112 pixels no formato PGM, são divididos em 40 diretórios um para cada indivíduo.



Figura 9 – Exemplo de imagens do ATT Face Database.

(CAMBRIDGE, 1994)

5.1.2 Extended Yale Facedatabase B

O Cropped Extended Yale Facedatabase B (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001), representado na Figura 10 contém 2470 imagens no total sendo 2432 de 38 pessoas (64 imagens por pessoa), as 38 imagens restantes são imagens do ambiente então foram descartadas neste trabalho, esta base de dados é uma versão menor e seu foco, é a extração de características que são resistentes contra a iluminação, essas imagens quase não possuem variações de emoções, nem outros elementos como barba, chapéis e variação da inclinação do rosto. O Extended Yale Facedatabase B é uma fusão do The Yale Facedatabase com o The Yale Facedatabase B. As imagens tem o tamanho de $168 \times 192pixels$, em preto e branco e formato .PGM, são divididos em 38 diretórios, um para cada indivíduo.



Figura 10 – Exemplo de imagens do Extended Yale Facedatabase B.

5.1.3 The Sheffield Face Database (The UMIST Face Database)

Essa base de dados, representado na Figura 11, contém originalmente 564 imagens de 20 pessoas diferentes, cada uma cobrindo uma variedade de poses de perfil frontal (GRAHAM; ALLINSON, 1998), excetuando-se esta, não há outras variações de imagens para uma mesma pessoa, como a utilização de óculos em uma imagem e a não utilização em outra. Para este trabalho utilizamos uma versão reduzida, 266 imagens de 14 indivíduos diferentes (19 imagens por indivíduo), as imagens possuem o tamanho $92 \times 112pixels$, em preto e branco (256 escalas de cinza), formato .PGM, e divididos em 19 diretórios um para cada indivíduo.

5.2 Métricas de Eficácia

As métricas de eficácia tem como objetivo avaliar os resultados gerados pelo sistema CBIR, inclusive no reconhecimento de faces como foi feito neste trabalho, é possível verificar qual foi a efetividade do nosso trabalho, nessa seção discutiremos três métricas de eficácia comumente usadas.



Figura 11 – Exemplo de imagens do The Sheffield Face Database.

5.2.1 Precisão

A métrica de eficácia Precisão é definida como a divisão de imagens relevantes pelo total de imagens recuperados em uma busca (FUJIKAWA; PEDRONETTE, 2014). Uma descrição muito utilizada para precisão é $P@n$, considerando os n primeiros elementos dentro da lista de resultados, ou seja que a precisão avalie até a n -ésima posição de um *ranked list*, a equação:

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Figura 12 – Equação que define a Precisão.

5.2.2 MAP (Mean Average Precision)

A *Average Precision* é a métrica mais comum para avaliar sistemas de recuperação de informações. Para se obter a precisão média de uma lista de resultados ranqueados, deve-se calcular a precisão toda vez que um objeto que é relevante ao contexto apareça na lista de resultados, em seguida calcula-se a média das precisões obtidas (FUJIKAWA; PEDRONETTE, 2014). Formalmente considerando q uma imagem de consulta, N_r o número de imagens relevantes à consulta q , considere $(r_i | i = 1, 2, \dots, d)$ um vetor que é ordenado pela relevância até uma profundidade d , sendo que r_i indica a relevância da i -ésima pontuação que o item alcançou, 0 sendo não relevante até 1 como relevante a precisão média é definida como:

$$AP = 1/N_r \sum_{i=1}^d (r_i/i \sum_{j=1}^i r_j) \quad (5.1)$$

O MAP é definido considerando a precisão média (AP) entre todas as imagens da coleção.

5.2.3 Recall (Revocação)

A revocação (DAVIS; GOADRICH, 2006) está relacionada à abrangência dos resultados recuperados, definida em termos da fração de instâncias pertinentes que são recuperadas. Seu cálculo é similar à métrica de precisão, é representada pela seguinte equação:

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Figura 13 – Equação que define a Revocação.

5.3 Resultados

Nesta seção serão discutidos os resultados que obtivemos neste trabalho. primeiramente daremos uma breve introdução sobre os parâmetros que utilizamos, e em seguida serão apresentados os resultados em tabelas e gráficos.

5.3.1 Métrica P@20; MAP

Como principais métricas para avaliação de eficácia deste trabalho utilizamos o MAP e o P@20. A métrica P@20 foi utilizada com o objetivo de não restringir muito a avaliação às primeiras posições e o MAP para possibilitar uma visão geral sobre a eficácia dos *ranked lists*. Em alguns cenários, consideramos também o P@10. Como podemos observar na medida MAP a base de dados ATT Facedatabase (CAMBRIDGE, 2002) obteve o melhor resultado dentre as 3. A grande divergência de acurácia entre o MAP e o P@20 se dá devido ao dataset que possui apenas 10 imagens por individuo, logo não sendo possível obter uma porcentagem acima de 50 %.

Para a métrica P@20 o maior valor foi alcançado no The Sheffield Face Database (GRAHAM; ALLINSON, 1998), com 62,05%, podemos notar que também obteve uma alta pontuação no MAP, ao contrário do Extended Yale Face Database B (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001), isso pode ser relacionado a grande quantidade de imagens, enquanto o Yale possui 2432 imagens (64 imagens por pessoa), o UMIST Face Database e o ATT Face Database ambas não ultrapassam 600 imagens cada uma.

5.3.2 Definição dos parâmetros

O parâmetro k representa quantos vizinhos próximos serão considerados no algoritmo de reclassificação iterativa RL-Sim*, nessa análise foi o parâmetro k foi variado

Tabela 1 – Resultados obtidos nas bases de dados sem a aplicação do RL-Sim*

Base de dados	MAP	P@20
ATT Face Database	71,08%	36,82%
Extended Yale Face Database B	8,54%	19,32%
The Sheffield Face Database	70,93%	62,05%

em tamanhos de 10,15 e 20, já o parâmetro T representa a quantidade de iterações do algoritmo RL-Sim* que foi variado entre os valores 1 e 3.

Primeiramente apresentaremos os resultados obtidos no ATT Face Database (CAMBRIDGE, 2002). Na métrica MAP obtivemos o melhor resultado em $k = 10$ e $T = 1$, o resultado foi de 78,6% um ganho total de +6,98% de acurácia. O pior resultado foi em $k=20$ e $T=3$, com um MAP de 69,58% que representa uma perda de -1,5% de acurácia. Essa perda se explica devido ao k ser maior que a quantidade possível de resultados certos possíveis, se agravando ainda mais reiterando com $T = 3$, que reitera por 3 vezes esse procedimento.

Tabela 2 – Resultados variando T e k utilizando o ATT Face Database MAP/P@20.

Métrica	k=10 T=1	k=10 T=3	k=15 T=1	k=15 T=3	k=20 T=1	k=20 T=3
MAP	78,60%	77,93%	77,09%	74,48%	75,10%	69,58%
P@20	40,50%	40,31%	39,91%	39,72%	39,14%	37,85%

Foram analisados também os resultados da métrica P@10. Podemos notar um alto ganho de eficácia, em $k=10$ e $T=3$ conseguimos uma pontuação de 73,72%, um ganho de +33,22%.

Tabela 3 – Resultados variando T e k utilizando o ATT Face Database P@10.

Métrica	Sem RL-Sim*	k=10 T=1	k=10 T=3	k=15 T=1	k=15 T=3	k=20 T=1
P@10	64,63%	73,72%	73,20%	72,13%	68,95%	69,85%

Aplicando o RL-Sim* nos resultados obtidos a partir do Extended Yale Face Database (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001), ambas maiores pontuações tanto no P@20 e no MAP foram obtidas com os parâmetros $k=10$ e $T=3$. Na métrica MAP obtivemos uma pontuação de 11,04% um ganho baixo de +2,5% entretanto alto em relação a baixa acurácia dos resultados obtidos anteriormente. Para o P@20 obtivemos um resultado de 25,01% um ganho de +5,69%, e a menor pontuação também assim como na base de dados anterior foi em $k=20$ e $T=3$ com 8,94% no MAP e 18,43%, respectivamente um ganho de +0,4% e uma perda de -0,89%.

Para o The Sheffield Face Database (GRAHAM; ALLINSON, 1998), assim como o Yale obtivemos os melhores resultados para os parâmetros de $k=10$ e $T=3$, obtivemos

Tabela 4 – Resultados variando T e k utilizando o Extended Yale Face Database B MAP/P@20

Métrica	k=10 T=1	k=10 T=3	k=15 T=1	k=15 T=3	k=20 T=1	k=20 T=3
MAP	9,91%	11,04%	9,46%	9,89%	8,98%	8,94%
P@20	23,29%	25,01%	21,03%	21,20%	19,33%	18,43%

um resultado de 86,61% para a métrica MAP e 80,53% para o P@20 um grande ganho de +15,68% para o MAP e de +18,48% para o P@20. Assim como para as outras base de dados, para o parâmetro k=20 e T=3 foram obtidos os piores resultados, mas mesmo assim superiores ao resultado sem o algoritmo RL-Sim*, obteve-se um ganho de +5,16% e +4,85% para o MAP e P@20 respectivamente.

Tabela 5 – Resultados variando T e k utilizando o The Sheffield Face Database MAP/P@20

Métrica	k=10 T=1	k=10 T=3	k=15 T=1	k=15 T=3	k=20 T=1	k=20 T=3
MAP	78,35%	86,61%	79,59%	83,36%	77,83%	76,09%
P@20	71,20%	80,53%	71,33%	74,57%	68,72%	66,90%

Nas Figuras 14 15 é possível visualizar os resultados retornados. as imagens quadriculadas em verde representam resultados da classe correta, na Figura 14 aparecem os resultados obtidos sem o RL-Sim*, na Figura 15 são os resultados após o RL-Sim* (OKADA; PEDRONETTE; TORRES, 2015)*, podemos visivelmente notar um grande aumento de acurácia com a aplicação do algoritmo.



Figura 14 – Recuperação sem o RL-Sim* @Sheffield Face Database.



Figura 15 – Recuperação com o RL-Sim* @Sheffield Face Database.

5.3.3 Curvas Precision x Recall

Nesta subseção apresentaremos os gráficos da curva de $Precision \times Recall$ variando os parâmetros T e k , essa curva nos fornece uma visão geral sobre os resultados da recuperação, uma vez que o Recall está associado a quão abrangente foram os resultados recuperados e a Precisão associado à qualidade dos resultados.

Analisando as Figuras 16 17, podemos notar que no início da curva não houve ganhos significativos. Os melhores resultados foram alcançados de modo geral em $k = 10$ e $T = 3$ em seguida do $k = 10$ e $T = 1$, isso pode ser explicado pelo ATT Face Database (CAMBRIDGE, 2002) possuir apenas 10 imagens por classe.

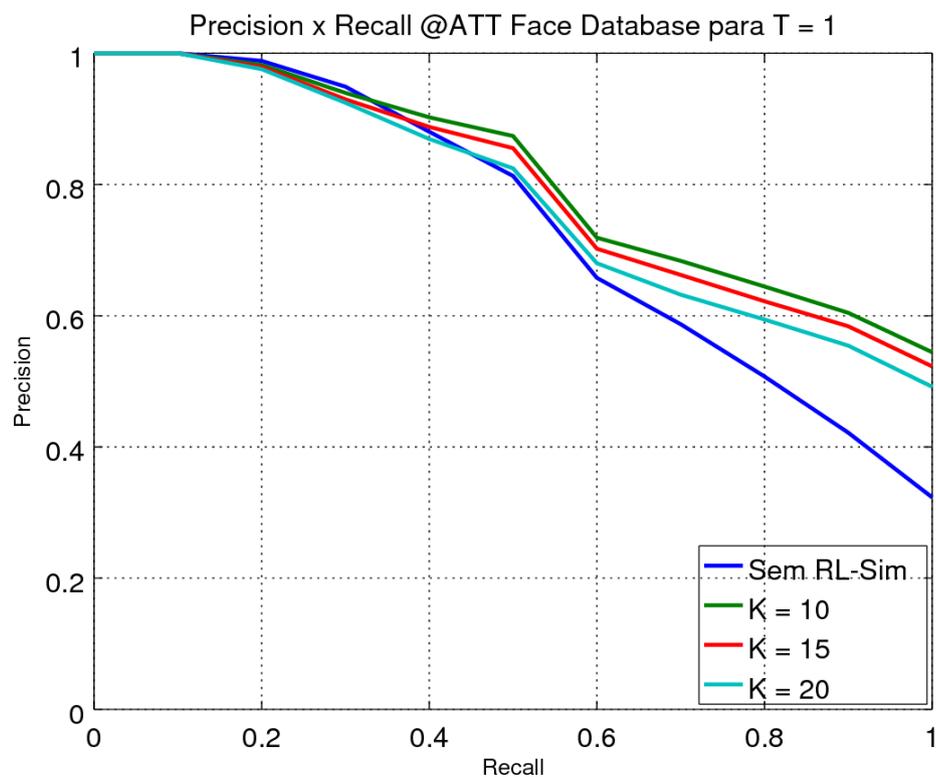


Figura 16 – Gráfico da curva $Precision \times Recall$ para o ATT Face Database.

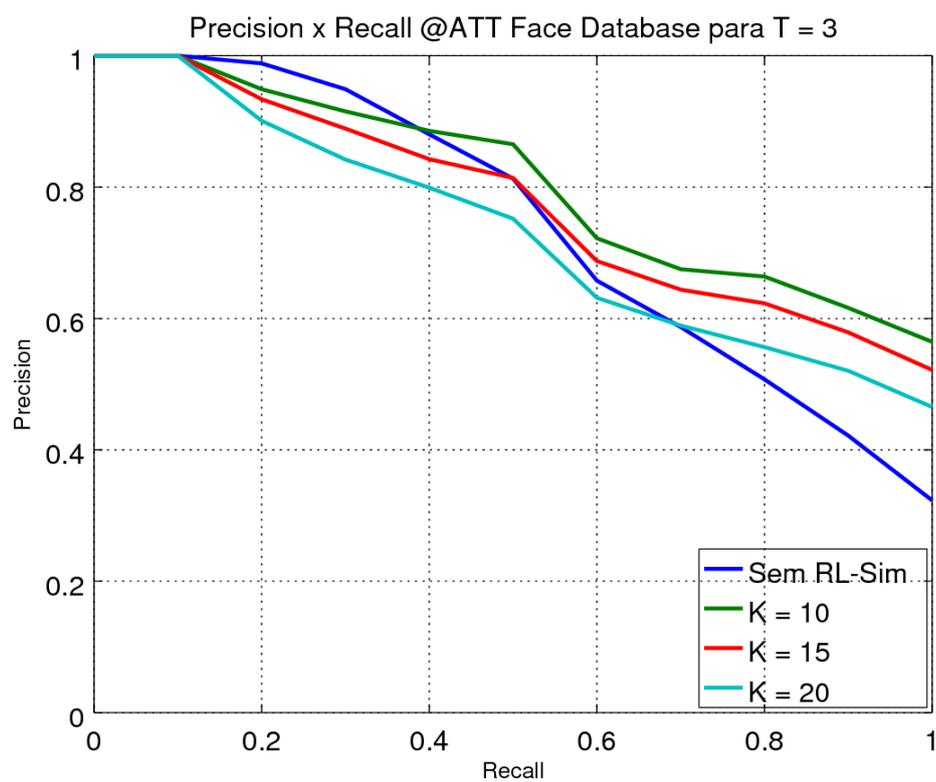


Figura 17 – Gráfico da curva $Precision \times Recall$ para o ATT Face Database.

Para as Figuras 18 19 podemos verificar que o início as curvas têm o início muito semelhante, assim como no fim e convergem para o mesmo ponto a partir do recall em 0.60, ou seja isso significa que não houve alterações a partir deste ponto com a aplicação do RL-Sim*. É interessante que os resultados foram baixos comparado com as outras base de dados, e também podemos visualizar que os melhores resultados não se concentraram em uma só coluna, eles vieram das três que possuem parâmetros diferentes, variando respectivamente $k = 10, 15$ e 20 .

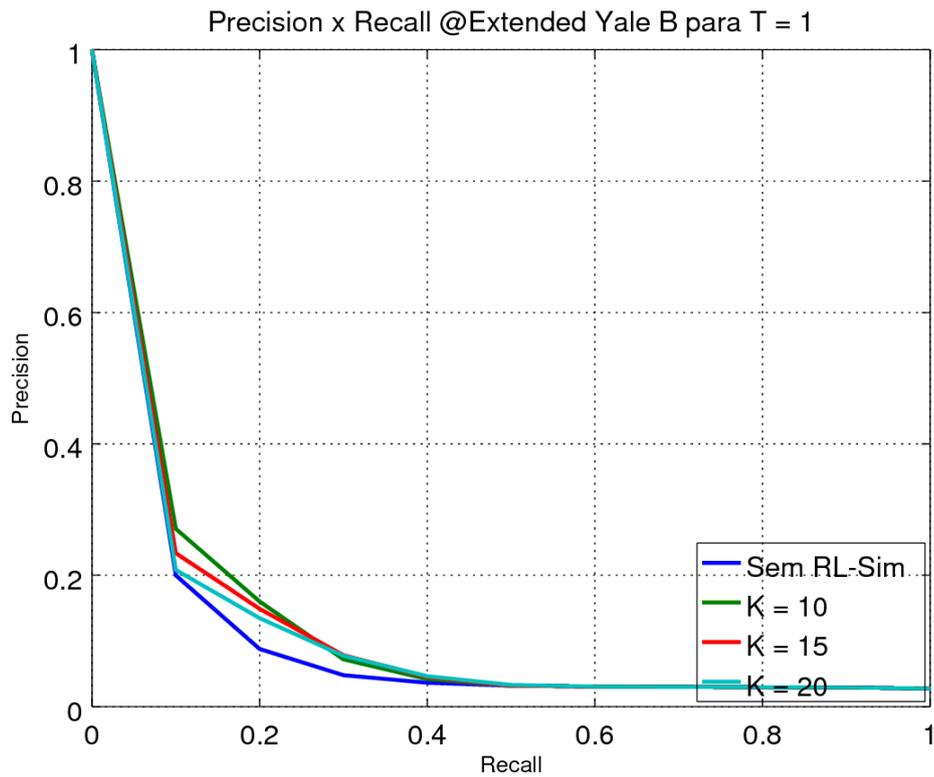


Figura 18 – Gráfico da curva $Precision \times Recall$ para o Extended Yale B.

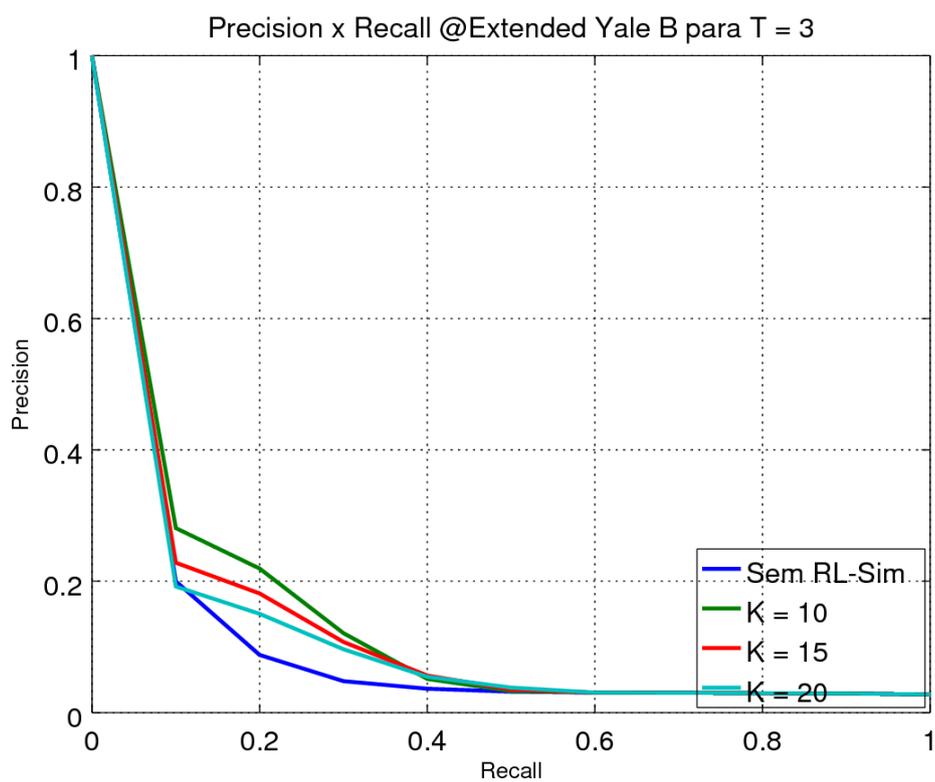


Figura 19 – Gráfico da curva $Precision \times Recall$ para o Extended Yale B.

Por fim, analisando as Figura 20 e 21 podemos notar visivelmente que para $T = 1$, o $k = 15$ obteve os melhores resultados e para $T = 3$, o $k = 10$ obteve os melhores que também no contexto geral teve os melhores ganhos. Podemos notar também que no início de ambas as figuras começam muito parecidas assim como nos outros dois datasets, e se aproximando ao fim os ganhos começam a ficar bem notáveis, muito similar ao ATT Face Dataset. Podemos verificar uma permanência dos resultados em $T = 3$ e $k = 10$, que também foram os melhores nas outras duas base de dados.

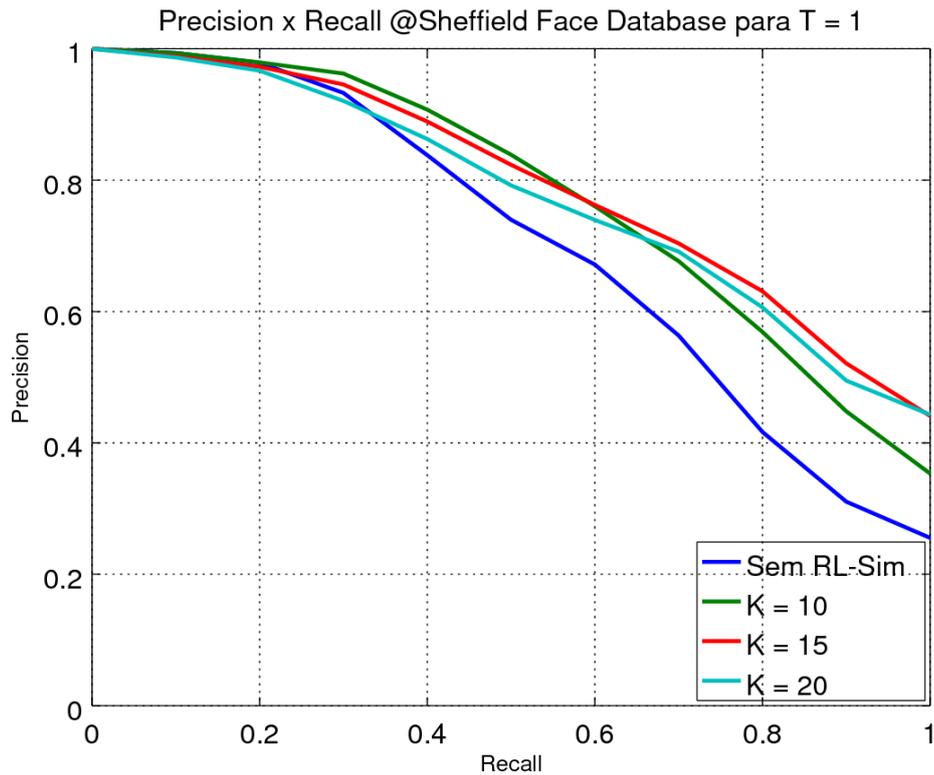


Figura 20 – Gráfico da curva $Precision \times Recall$ para o UMIST Face Database.

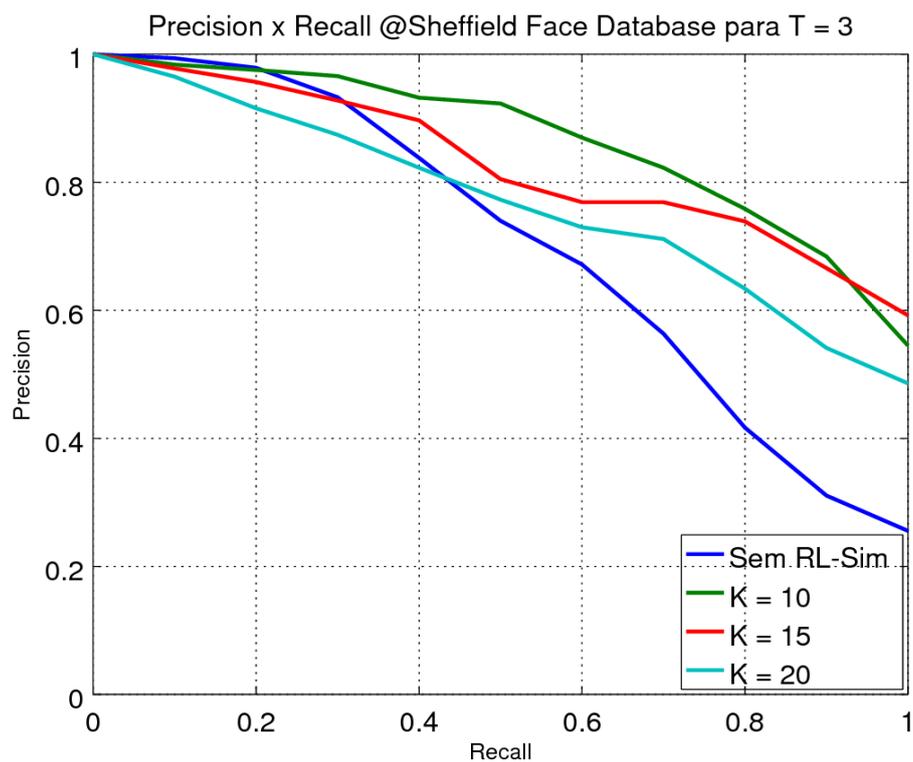


Figura 21 – Gráfico da curva *Precision x Recall* para o UMIST Face Database.

6 Conclusão

Neste trabalho apresentamos uma série de técnicas que abordam o reconhecimento de face, desde extração de vetores, até sua classificação com alguma métrica de distância, também foi aplicada técnicas pós processamento não supervisionadas que tem por objetivo fazer uma reclassificação iterativa, a partir dos resultados gerados pelas métricas de distâncias.

Na prática escolhemos o Local Binary Patterns (AHONEN; HADID; PIETIKAINEN, 2006), um descritor de textura, para realizar a tarefa de extração de vetores em imagens de rostos humanos, esse algoritmo nos retorna uma matriz de características que posteriormente foi aplicado o cálculo de distância Euclidiana, gerando uma lista de resultados contendo as faces mais similares, com essa classificação aplicamos o RL-Sim* (OKADA; PEDRONETTE; TORRES, 2015) para tentar melhorar o ranqueamento obtido pelas distâncias Euclidianas.

Concluimos que o reconhecimento de faces não é uma tarefa trivial, uma vez que os resultados obtidos inicialmente utilizando apenas as distâncias não foi uma recuperação eficaz. Entretanto também podemos concluir que foi e é possível aumentar significamente a acurácia de abordagens de reconhecimento de faces utilizando técnicas de pós processamento de aprendizado não supervisionado. Obtivemos alguns resultados com aproximadamente 20% de ganho em algumas métricas como o MAP. Em relação as curvas de *Precision* \times *Recall*, podemos perceber um ganho significativo, principalmente nas base de dados com uma quantidade menor de imagens.

Durante o trabalho foi possível entender como o reconhecimento facial é realizada e como pode ser melhorado, e ao mesmo tempo verificar a atuação do algoritmo RL-Sim* dentro desse contexto, contribuindo para uma utilização mais efetiva.

Referências

- AHONEN, T.; HADID, A.; PIETIKAINEN, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 28, n. 12, p. 2037–2041, dez. 2006. ISSN 0162-8828. Disponível em: <<http://dx.doi.org/10.1109/TPAMI.2006.244>>. Citado 4 vezes nas páginas 7, 19, 25 e 37.
- BARAHATE, S. R.; SATURWAR, J. Face recognition using pca based algorithm and neural network. In: *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*. New York, NY, USA: ACM, 2010. (ICWET '10), p. 249–252. ISBN 978-1-60558-812-4. Disponível em: <<http://doi.acm.org/10.1145/1741906.1741963>>. Citado 2 vezes nas páginas 9 e 10.
- BELHUMEUR, P. N.; HESPANHA, J. a. P.; KRIEGMAN, D. J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 19, n. 7, p. 711–720, jul. 1997. ISSN 0162-8828. Disponível em: <<http://dx.doi.org/10.1109/34.598228>>. Citado 2 vezes nas páginas 9 e 12.
- CAMBRIDGE. *The Database of Faces at a glance*. 1994. Disponível em: <<http://www.cl.cam.ac.uk/research/dtg/attarchive/facesataglance.html>>. Citado na página 25.
- CAMBRIDGE. *ATT Laboratories Cambridge database of faces*. 2002. Disponível em: <<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>>. Citado 4 vezes nas páginas 25, 28, 29 e 31.
- CARNEIRO, L. N. das V. Reconhecimento de face invariante a iluminação baseado em uma abordagem supervisionada. In: . [S.l.: s.n.], 2012. Citado na página 6.
- DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: ACM, 2006. (ICML '06), p. 233–240. ISBN 1-59593-383-2. Disponível em: <<http://doi.acm.org/10.1145/1143844.1143874>>. Citado na página 28.
- EDINBURGH. *PCA Example: Eigen Faces*. 2014. Disponível em: <<http://homepages.inf.ed.ac.uk/vlavrenk/iaml.html>>. Citado na página 10.
- FALCAO, A. X.; TORRES, R. da S. Content-based image retrieval: Theory and applications. In: *RITA*. [S.l.: s.n.], 2006. Citado na página 17.
- FUJIKAWA, C. S.; PEDRONETTE, D. C. G. Análise e combinação de medidas não-supervisionadas de estimativa de eficácia. In: *Workshop of Undergraduate Work - 27th SIBGRAPI: Conference on Graphics, Patterns and Images*. [S.l.: s.n.], 2014. Citado 4 vezes nas páginas 6, 17, 18 e 27.
- GEORGHIADES, A. S.; BELHUMEUR, P. N.; KRIEGMAN, D. J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington,

- DC, USA, v. 23, n. 6, p. 643–660, jun. 2001. ISSN 0162-8828. Disponível em: <<http://dx.doi.org/10.1109/34.927464>>. Citado 4 vezes nas páginas 25, 26, 28 e 29.
- GRAHAM, D. B.; ALLINSON, N. M. *The UMIST Database*. 1998. Disponível em: <<https://www.sheffield.ac.uk/eee/research/iel/research/face>>. Citado 4 vezes nas páginas 25, 26, 28 e 29.
- OKADA, C. Y.; PEDRONETTE, D. C. G.; TORRES, R. da S. Unsupervised distance learning by rank correlation measures for image retrieval. In: *CM International Conference on Multimedia Retrieval (ICMR'2015)*. [S.l.: s.n.], 2015. Citado 4 vezes nas páginas 7, 23, 30 e 37.
- PARDESHI, M. *High Dimension CNF to DNF Conversion using GRID Computing*. 2013. Disponível em: <<https://mayuresha.wordpress.com/page/3/>>. Citado na página 15.
- PEDRONETTE, D. C. G.; TORRES, R. da S. Exploiting contextual information for image re-ranking and rank aggregation. *International Journal of Multimedia Information Retrieval*, v. 1, n. 2, p. 115–128, 2012. Citado 2 vezes nas páginas 7 e 21.
- PEDRONETTE, D. C. G.; TORRES, R. da S. Unsupervised measures for estimating the effectiveness of image retrieval systems. In: *26th SIBGRAPI - Conference on Graphics, Patterns and Images*. [S.l.: s.n.], 2013. p. 341–348. Nenhuma citação no texto.
- PEDRONETTE, D. C. G. et al. Rl-sim algorithm acceleration on GPUs. In: *SBAC*. [S.l.: s.n.], 2013. Citado 3 vezes nas páginas 21, 22 e 23.
- PENATTI, A. B.; VALLE, E.; TORRES, R. da S. Comparative study of global color and texture descriptors for web image retrieval. In: *Journal of Visual Communication and Image Representation*. [S.l.: s.n.], 2012. Citado na página 19.
- RADY, H. Face recognition using principle component analysis with different distance classifiers. In: *JCSNS International Journal of Computer Science and Network Security*. [S.l.: s.n.], 2011. Citado 2 vezes nas páginas 15 e 16.
- SILVA, A. T. *Descritores de Imagem*. 2016. Disponível em: <http://www.joinville.udesc.br/portal/professores/andretavares/materiais/RP_Aula05_Descritores.pdf>. Citado 2 vezes nas páginas 17 e 18.
- SOHAIL, A. S. M.; BHATTACHARYA, P. Classification of facial expressions using k-nearest neighbor classifier. In: *Proceedings of the 3rd International Conference on Computer Vision/Computer Graphics Collaboration Techniques*. Berlin, Heidelberg: Springer-Verlag, 2007. (MIRAGE'07), p. 555–566. ISBN 978-3-540-71456-9. Disponível em: <<http://dl.acm.org/citation.cfm?id=1759437.1759489>>. Citado na página 14.
- TRAN, M. *Eigenface for Face Recognition Presentation*. 2008. Disponível em: <http://www.slideshare.net/minhkiller/eigenface-for-face-recognition-presentation/18-ulliDemo_liul>. Citado na página 12.
- TURK, M.; PENTLAND, A. Eigenfaces for recognition. *J. Cognitive Neuroscience*, MIT Press, Cambridge, MA, USA, v. 3, n. 1, p. 71–86, jan. 1991. ISSN 0898-929X. Disponível em: <<http://dx.doi.org/10.1162/jocn.1991.3.1.71>>. Citado 3 vezes nas páginas 9, 10 e 12.

ZHAO, W. et al. Face recognition: A literature survey. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 35, n. 4, p. 399–458, dez. 2003. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/954339.954342>>. Citado na página 14.