

RedFace: Um Sistema de Reconhecimento Facial Baseado em Técnicas de Análise de Componentes Principais e Autofaces

Fábio A. Diniz¹, Francisco Milton Mendes Neto¹, Laysa Mabel de O. Fontes¹.

fabio.abrantes.diniz@gmail.com, {miltonmendes, laysa}@ufersa.edu.br

¹Programa de Pós-Graduação em Ciência da Computação – PPgCC
Universidade do Estado do Rio Grande do Norte – UERN
Universidade Federal Rural do Semi-Árido – UFERSA
BR 110 – Km 46 – Bairro Costa e Silva – Campus Central
59.625-620 Mossoró – RN – Brasil

Resumo: *Desenvolver um modelo computacional de reconhecimento facial não é uma tarefa trivial, porque as faces e os estímulos visuais multidimensionais possuem características de modelagem complexa. A grande dificuldade está na modelagem de uma face que abstraia as características que as diferenciem de outras faces, já que estas apresentam poucas diferenças substanciais entre si. Embora diferentes, todas as faces possuem características como, por exemplo, uma boca, dois olhos e um nariz. No presente trabalho é proposto um sistema de reconhecimento facial desenvolvido em duas fases. Inicialmente utilizam-se as técnicas de Análise de Componentes Principais (PCA) e Eigenfaces (autofaces) para a extração de características da face. Na segunda fase foram aplicados os classificadores K-Nearest Neighbors (K-NN), Random Forest (Floresta Aleatória) e K-Star (K-estrela) no processo de reconhecimento da face. A validação dos algoritmos foi realizada numa base de dados contendo 1280 imagens de 64 classes distintas. Finalmente, foi mostrado que o desempenho dos algoritmos testados para sistemas de reconhecimentos de face baseado em PCA foram muito satisfatórios, atingindo as melhores taxas de reconhecimento, acima de 90% em todos os classificadores.*

Abstract: *To develop a computational model of face recognition is not a trivial task, because the faces and the visual characteristics are multidimensional modeling complex. The great difficulty lies in modeling a face that abstracts the characteristics that differentiate it from other faces, as they have few substantial differences between them. While different, all the faces have characteristics such as a mouth, two eyes and nose. In this paper we propose a face recognition system developed in two phases. Initially using the techniques of Principal Component Analysis (PCA) and Eigenfaces for extracting features of the face. In the second phase were applied classifiers K-Nearest Neighbors (K-NN), Random Forest and K-Star in the process of face recognition. Validation algorithm was conducted on a database 1280 containing 64 different classes of images. Finally, it was shown that the performance of the proposed algorithms for face recognition systems based on PCA were very good, reaching the best recognition rates above 90% in all binders.*

Palavras-chave: Análise de Componentes Principais, K-Nearest Neighbors, Random Forest, k-Star, Autofaces.

1. Introdução

O reconhecimento facial é um dos processos de identificação mais utilizados pelos seres humanos, pois permite identificar rapidamente qualquer indivíduo. Além de identificar, podemos perceber o estado emocional de uma pessoa apenas observando sua expressão facial [Ekman, 1993]. Embora o reconhecimento facial seja uma tarefa simples para o ser humano, não é trivial implementar esse processo em uma máquina, pois não se sabe, ao certo, como o cérebro humano realiza essa tarefa [Agarwal et al, 2010].

Diferentes técnicas serão utilizadas, neste trabalho, para a construção de um sistema de reconhecimento facial, denominado RedFace. A partir de uma imagem dinâmica, capturada pela *webcam* do usuário, o sistema irá extrair a imagem facial de um indivíduo e verificar se o mesmo pertence ou não a uma base de dados com imagens faciais previamente cadastradas. Este trabalho utilizou as técnicas de Análise de Componentes Principais (PCA) e Eigenfaces (autofaces) [Stan;Anil, 2011] no processo de extração de características da face para diminuir a alta dimensionalidade dos vetores [Stan;Anil, 2011]. A classificação das imagens foi realizada através de análises dos classificadores K-Nearest Neighbors (K-NN) [Jiangsheng, 2002], Random Forest (Floresta aleatória)

[Breiman, 2001] e K-Star (K-estrela) [Cleary;Trigg, 1995]. O propósito foi descobrir qual combinação destas técnicas tem melhor desempenho em termos de taxa de reconhecimento facial.

O presente trabalho está organizado como segue. Seção 2 discute os trabalhos relacionados na área de sistema de reconhecimento facial. A Seção 3 apresenta a descrição do sistema de reconhecimento facial proposto, mostrando as suas etapas de implementação. A Seção 4 descreve os métodos PCA e Eigenface. A Seção 5 apresenta uma análise dos resultados. Por fim, as conclusões finais e trabalhos futuros são apresentados na Seção 6.

2. Trabalhos Relacionados

Existem duas abordagens básicas para o reconhecimento da face [Stan;Anil, 2011]. A primeira abordagem é baseada na extração de vetores característicos de partes básicas de uma face, tais como: olhos, nariz, boca e queixo. Esta abordagem utiliza modelos deformáveis e análise matemática para extrair informação das componentes básicas da face e, em seguida, converter essa informação em um vetor característico [Stan;Anil, 2011]. A segunda abordagem é baseada nos conceitos da Teoria da Informação [Stan;Anil, 2011]. Nesta abordagem

a informação que melhor descreve uma face é derivada a partir da imagem da face toda.

Geralmente, modelos deformáveis de face visando extrair características usam retângulos para localizar os principais elementos da face (tais como boca, olhos e sobrancelhas) e, em seguida, são projetados modelos de curvas e funções de energia que caracterizam olhos, boca, queixo e contorno da face [Zhang; Ruan, 2006].

Em [Ghosal et al, 2009], a extração das características de imagens da face pode ser obtida usando análise matemática com a transformada de *Wavelet Gabor*. Tais características podem ser obtidas através do algoritmo *Random Forest*, classificador de imagens da face com base nas características extraídas. O uso da transformada *Wavelet Gabor* conduz a um número extremamente elevado de características, a maioria das quais são redundantes para efeitos de classificação. Tal problema pode ser contornado eficientemente usando a técnica de *Random Forest* para calcular as características mais importantes de *Wavelet Gabor*, uma vez que o espaço das características é reduzido, acelerando o processo da classificação facial.

Em [Kshirsagar et al, 2011], os métodos de análise dos componentes principais (PCAs) e os métodos estatísticos foram utilizados. Em [Stan;Anil, 2011], foi implementado um sistema de reconhecimento de face utilizando as

técnicas de Eigenfaces, PCA e Redes Neurais. Neste trabalho foi usado o algoritmo PCA, o qual extrai as características mais relevantes que representam uma identidade única da imagem, características usadas como entradas da rede neural classificadora da face. Foi comprovado que os Eigenfaces têm a capacidade de fornecer as características significativas e reduz o tamanho de entradas da rede neural, aumentando a velocidade no processo de reconhecimento. No entanto, esta abordagem é sensível a imagens com condições de iluminação não controladas.

Com base no exposto, este trabalho consiste em projetar um sistema de reconhecimento facial (RedFace) utilizando, inicialmente, as técnicas de Análise de Componentes Principais (PCA) e Eigenfaces (autofaces) para a extração de características da face e, em seguida, aplicar os classificadores *K-Nearest Neighbors* (K-NN), *Random Forest* e *K-Star* na classificação das faces.

3. Sistemas de Reconhecimento Facial

O sistema de reconhecimento facial proposto consiste basicamente das seguintes etapas: (a) aquisição das imagens; (b) pré-processamento; (c) extração de características; (d) classificação e verificação [Stan;Anil, 2011]. A arquitetura do RedFace é apresentada na Figura 1.

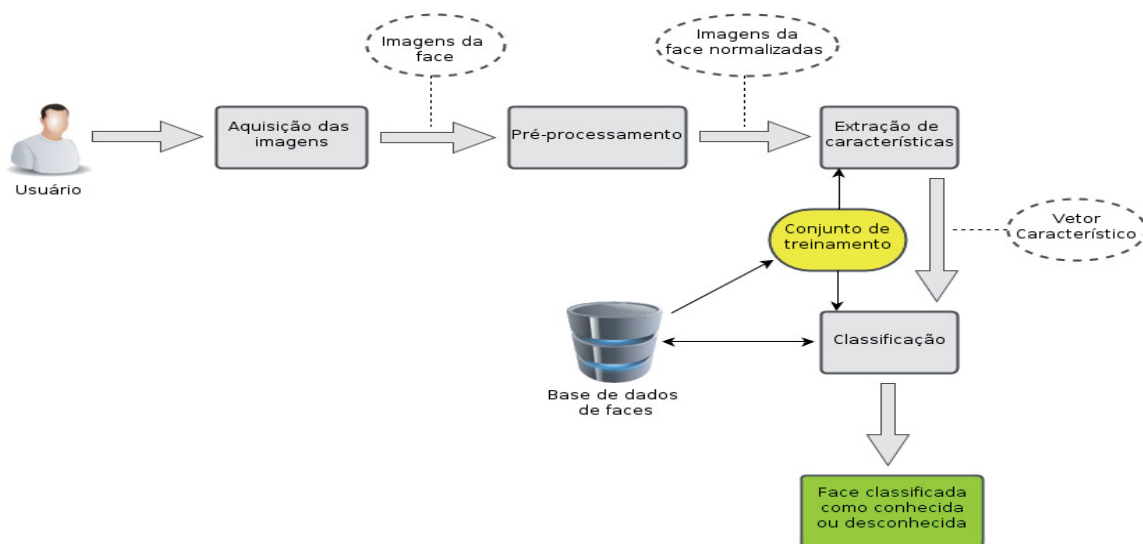


Figura 1. Arquitetura do Sistema de Reconhecimento Facial.

O início do processo de reconhecimento de face se dá a partir do módulo de aquisição da imagem. Através da *webcam*, o sistema captura a imagem da face do usuário a qual será utilizada como entrada para o sistema de reconhecimento facial. O próximo módulo é o pré-processamento da imagem. Neste módulo as imagens são normalizadas e corrigidas para melhorar o reconhecimento da face. As fotos tiveram correções na dimensão (todas foram passadas para 120x90, 32x24 e 12x9), na resolução (300 dpi) e na iluminação (foi utilizada a mesma em todas as fotos) e foram passadas para escala de cinza.

Após o pré-processamento, a imagem da face normalizada serviu de entrada para o módulo de extração de características, objetivando encontrar as principais

características a serem utilizadas para a classificação. Cada imagem representante do indivíduo foi transformada em uma matriz de tamanho $w \times h$, onde w e h são, respectivamente, os números de pixels referentes à largura e altura da imagem [Romdhani, 1996]. Cada valor do pixel da imagem corresponde a uma componente do vetor.

Devido à alta dimensionalidade dos vetores, foi utilizada, nesta etapa, a técnica de PCA, visando reduzir a dimensionalidade de uma imagem a fim de diminuir o custo computacional e melhorar a precisão do classificador. Se o espaço característico contiver somente as características mais visíveis, o classificador será mais rápido e ocupará menos memória [Agarwal et al, 2010]. Juntamente com a técnica PCA, foi utilizada a técnica do

Eigenfaces. O algoritmo Eigenface objetiva fornecer um conjunto de vetores de distribuições probabilísticas para resolver o problema da detecção de padrões em imagens e seu fundamento básico é a utilização desses vetores para gerar uma informação matemática do rosto de um ser humano para sua futura identificação.

Com a ajuda de um classificador de padrões, as características extraídas da imagem da face são comparadas com as amostras do conjunto de treinamento de face. A imagem da face é então classificada como conhecida ou desconhecida. Neste trabalho foram analisados três algoritmos de reconhecimento de padrões: K-NN [Jiangsheng, 2002], *Random Forest* [Breiman, 2001] e *K-Star* [Cleary;Trigg, 1995] detalhadas nas subseções seguintes. Estes dois últimos foram utilizados a partir do software de mineração de dados Weka (*Waikato Environment for Knowledge Analysis*) [Weka, 2011].

O outro módulo da Figura 1 é o conjunto de treinamento. Ele visa encontrar características apropriadas para a representação de padrões de entrada, sendo o classificador treinado para particionar o espaço de características, formando padrões de todas as classes existentes. O último componente visto na Figura 1 que este trabalho utilizou foi uma base de dados própria, possuindo 1280 imagens de um total de 64 pessoas. A Figura 2 ilustra algumas faces armazenadas na base de dados utilizada.



Figura 2. Imagens de faces da base de dados.

Cada indivíduo integrante da base de dados é representado por 20 imagens nas seguintes posições: 4 imagens olhando diretamente para a câmera, 4 imagens olhando 45 graus à esquerda da câmera, 4 imagens olhando 45 graus à direita da câmera, 4 imagens olhando 90 graus à esquerda da câmera e 4 imagens olhando 90 graus à direita da câmera, como pode ser visto na Figura 3. Dessas 20 imagens, 5 (cada uma de uma posição) compõem o conjunto de treinamento do sistema e as 15 restantes para o conjunto de teste.



Figura 3. Esquema de posicionamento da face.

3.1. Classificador *K-Nearest Neighbors*

O K-NN é um dos algoritmos de classificação mais utilizados na área de aprendizagem de máquina. É baseado na procura dos k vizinhos mais próximos do padrão de teste [Jiangsheng, 2002]. A busca pela vizinhança é feita utilizando uma medida de distância nessa procura. Nos experimentos desenvolvidos neste trabalho, além da medida euclidiana, foram utilizadas: a distância de Manhattan, e a euclidiana normalizada [Jiangsheng, 2002]. Além disso, todos os experimentos com K-NN foram realizados com peso pela distância e não pela frequência. Dessa maneira, o padrão é classificado de acordo com a soma dos pesos dos k vizinhos – o peso é o inverso das distâncias. O padrão é dito pertencer à classe que apresentar maior frequência dentre os k vizinhos utilizados [Jiangsheng, 2002]. O k escolhido será o que proporcionar menor quantidade erros na classificação das imagens.

3.2. Classificador *Random Forest*

O método de classificação conhecido como *Random Forests* foi proposto em [Breiman, 2001]. Ele consiste em uma técnica de agregação de classificadores do tipo árvore, construídos de forma que a sua estrutura seja composta de maneira aleatória. Para determinar a classe de uma instância, o método combina o resultado de várias árvores de decisão, por meio de um mecanismo de votação. Cada árvore dá uma classificação, ou um voto para uma classe. A classificação final é dada pela classe que recebeu o maior número de votos entre todas as árvores da floresta [Breiman, 2001].

O classificador é baseado no método *Bagging* [Breiman, 2001]. Para cada árvore gerada é utilizado um conjunto de treinamento diferente, formado por n instâncias de treinamento escolhidas aleatoriamente (i.e., uma amostra ou *bootstrap*). Para cada nó da árvore gerada, são escolhidos aleatoriamente m atributos que orientam o direcionamento do nó, baseado na melhor discriminação de classes do conjunto de treinamento, de acordo com uma métrica. Em geral, o valor de m deve ser bem menor que o total de atributos da base, de maneira que possam ser geradas árvores distintas, que são combinadas para classificar uma nova instância [Breiman, 2001].

O modelo gerado elege a classe mais frequente entre as opções individuais de cada árvore. Dessa forma, a seleção

de atributos é feita no instante de construção do modelo de classificação, caracterizando a seleção do tipo embutida (*embedded*). A vantagem desse classificador é que ele permite bases de dados com um número grande de atributos, contudo é suscetível a super ajuste (*overfitting*) em determinadas bases [Segal, 2004].

Como mencionado anteriormente, esse algoritmo foi utilizado a partir da ferramenta *Weka*, que proporciona a variação do número de árvores e o número de características a serem analisadas. Escolhemos essa ferramenta por ser amplamente utilizada em atividades de aprendizado de máquina e por ser adequada a conjunto dados ruidosos, apresentando um bom desempenho na análise de imagens [Zainal; Maarof, 2008].

3.3. Classificador *K-Star*

O *K-Star* pertence à mesma família que o K-NN, ou seja, também faz uso dos k vizinhos mais próximos [Cleary; Trigg, 1995]. O *K-Star* também foi executado no *Weka*, variando o parâmetro *Global Blend* (Combinação Global), definido em porcentagem [Weka, 2011]. Quando esse parâmetro é igual a zero, o *K-Star* se comporta exatamente como o K-NN [Cleary; Trigg, 1995].

K-Star é um classificador baseado em exemplos, isto é, baseia-se na classe das instâncias de formação semelhante, conforme determinado por uma função de similaridade [Cleary; Trigg, 1995]. Ele difere de outros por utilizar funções de distância baseadas na entropia e assume que os exemplos similares terão classes similares. O algoritmo *K-Star* utilizado consiste em uma implementação em Java, sendo parte integrante do pacote *Weka* [Frank *et al.*, 1999]. O *K-Star* possui os parâmetros de configuração para a execução os quais são *globalBlend* e *missingMode* [Weka, 2011].

O *K-Star* possui os parâmetros de configuração para a execução são [Frank *et al.*, 1999][Weka, 2011]:

- **Global Blend:** Parâmetro correspondente à mistura, relacionado ao uso de cálculos de entropia, com valores definidos de 0 a 100 e expressos em porcentagem (B=20);
- **Missing Mode:** Determina como os valores de atributos faltantes são tratados.

4. Descrição dos Métodos

Nessa seção, serão detalhados os métodos PCA e *Eigenface*, ambos abordados, neste trabalho, para realização do reconhecimento facial.

4.1. PCA

A análise da componente principal é uma técnica matemática que descreve um conjunto de dados usando “componentes principais”, escrita como combinações lineares dos dados originais. As componentes principais são determinadas em ordem decrescente de importância. A primeira componente possui mais informação do que a segunda, e assim por diante. O PCA tenta construir um pequeno conjunto de componentes que resumem os dados originais, reduzindo a dimensionalidade dos mesmos,

preservando os componentes mais significantes [Kshirsagar *et al.*, 2011].

As componentes principais, consideradas as mais importantes, possuem as direções em que os dados de entrada têm as maiores variâncias e as outras componentes (componentes com as menores variâncias, chamadas de componentes menores) são consideradas menos importantes ou associadas a ruídos.

O PCA é baseado nos autovetores e autovalores da matriz covariância dos dados [Agarwal *et al.*, 2010]. O objetivo é encontrar um conjunto de vetores ortonormais v_i que melhor descreve a distribuição dos dados de entrada. Seja C a matriz covariância estimada:

$$C = XX^T = VAV^T \in R^{n \times n} \quad (1)$$

Em que $A = \text{diag} \{ \lambda_1, \dots, \lambda_n \}$ é a matriz diagonal contendo n autovalores, v_i é o autovetor associado a λ_i e $V = [v_1, \dots, v_n] \in R^{n \times n}$ é a matriz dos autovetores, v_i , da matriz covariância

$$X = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nm} \end{bmatrix} \quad (2)$$

Onde X é a matriz de dados originais, em que $x_{ij} \in R$, $i = 1, \dots, n$ e $j = 1, \dots, m$.

O PCA determina uma transformação linear dos elementos de X , tal que:

$$y = V^T X \quad (3)$$

Onde $y = [y_1, y_2, \dots, y_n]$ é o vetor das componentes principais.

4.2. Eigenface

O método *Eigenfaces* busca um conjunto de características que não depende das formas geométricas da face (olhos, nariz, orelhas e boca), utilizando toda a informação da representação facial [Kshirsagar *et al.*, 2011]. Seu funcionamento é similar ao funcionamento do PCA, entretanto é utilizada uma leve otimização para reduzir a matriz de covariância, reduzindo o processamento necessário para fazer o cálculo de seus autovetores e autovalores.

Baseadas na Teoria da Informação, as *Eigenfaces* buscam identificar um pequeno número de características que são relevantes para diferenciar uma face de outras faces. Essas características podem ser analisadas apenas com a variação dos valores assumidos pelos *pixels*, em um conjunto de imagens de faces.

Os autovetores da matriz de covariância de um conjunto de imagens de faces descreve a variação dos *pixels* em uma região diferente das imagens, ou seja, significa que cada autovetor descreve a variação dos *pixels* associados a diferentes características faciais [Turk, 1991].

Depois de encontrar os autovetores, as imagens de faces são projetadas em um espaço de características (“espaço de faces”) que melhor representa a variação entre faces conhecidas. Este espaço é definido pelas *Eigenfaces*, que

são os autovetores do conjunto de faces. O reconhecimento de uma face é realizado pela sua projeção no subespaço gerado pelas *Eigenfaces* e então pela comparação da posição obtida com a posição de indivíduos conhecidos [Turk, 1991].

A sequência dos procedimentos da geração das *Eigenfaces* abordados por esse trabalho foi [26]:

- I. Adquirir um conjunto de M faces. Onde $\Gamma_i (i \in N)$ é cada face do conjunto da base de treinamento.

$$\Gamma = [\Gamma_1, \Gamma_2, \dots, \Gamma_M] \quad (1)$$

- II. Calcular a face média $\Psi(2)$, para eliminar muita informação redundante na face, isto é, uma imagem de face poderia ser representada com poucas componentes principais devido à sua redundância.

$$\Psi = \frac{\sum_{i=1}^M \Gamma_i}{M} \quad (2)$$

- III. Criar uma matriz com as faces de treino com os *pixels* dispostos em linhas e as M faces do conjunto de treinamento dispostas em colunas.
- IV. Subtrair a imagem média Ψ de cada imagem de Γ , gerando uma nova matriz A (4) que contém somente as variações de cada face em relação à face média. O vetor Φ (3) contém todas as variações de uma determinada face Γ em relação à face média Ψ .

$$\Phi = \Gamma - \Psi \quad (3)$$

$$A = [\Phi_1, \Phi_2, \dots, \Phi_m] \quad (4)$$

- V. Sendo M menor que a dimensionalidade (largura * altura das imagens de treinamento) das imagens em Φ , calcular a matriz de covariância C (5).

$$C = AA^T \quad (5)$$

- VI. Calcular os autovetores v e autovalores λ da matriz C .
- VII. Criar a matriz de transformação. A matriz μ (6) conterá $M - 1$ autovetores significativos. Entretanto ainda é possível realizar a eliminação de alguns desses $M-1$ autovetores pela ordem de importância de seus correspondentes autovalores λ , gerando M' autovetores escolhidos.

$$\mu = v\Phi \quad (6)$$

- VIII. Normalizar os M' vetores da matriz μ .

- IX. As imagens do conjunto de treinamento são projetadas no espaço de faces, efetuando-se a operação de transformação (7). Em seguida o novo padrão passa para a fase de classificação.

$$\Omega = \mu^T \Phi \quad (7)$$

Figura 4 apresenta a face média do conjunto de treinamento (Figura 2) utilizado neste trabalho e algumas *Eigenfaces* calculadas para o conjunto de imagens de treino de exemplo.



Figura 4. Exemplo de face média (acima) e *Eigenfaces* mais importantes para o conjunto de treino.

5. Experimentos e Resultados

Os experimentos foram conduzidos com a mudança das técnicas e dos seus parâmetros. Todos os algoritmos e técnicas (assim como suas variações) foram executados de forma automática e exaustiva, de modo que foi analisado somente o melhor resultado da taxa de acerto para o conjunto de testes (número de acertos dividido pelo número total de testes), utilizando o 1-NN (primeiro vizinho mais próximo) e distância euclidiana normalizada. Primeiro foi executado o redimensionamento (*resize*) das imagens, resultando em três dimensões diferentes: 120x90 – tamanho normal das imagens; 32x24 – tamanho aproximado de um ícone [Hyun-Chul; Se-Young, 2012] e 12x9 – melhor resultado dos testes automáticos.

Em seguida, os testes foram desenvolvidos tentando encontrar o melhor número de características restantes após o PCA (número de autovetores do Eigenfaces). Foram utilizados valores de 15 a 120, e em caso de empate (mesma taxa de acerto para número de características diferentes), o menor número de características foi escolhido como melhor por garantir menor custo computacional. O resultado obtido para cada dimensão foi: 120x90 – 17 características; 32x24 – 19 características e 12x9 – 20 características.

A Figura 5 a seguir mostra os resultados da taxa de acerto segundo as configurações: as três dimensões das imagens; número de características variando de 15 a 20 (não são mostrados os testes completos por serem muito extensos e por não terem apresentado bons resultados); e k no intervalo de 1 a 10 (k -NN); fazendo uso das três medidas de distância já citadas.

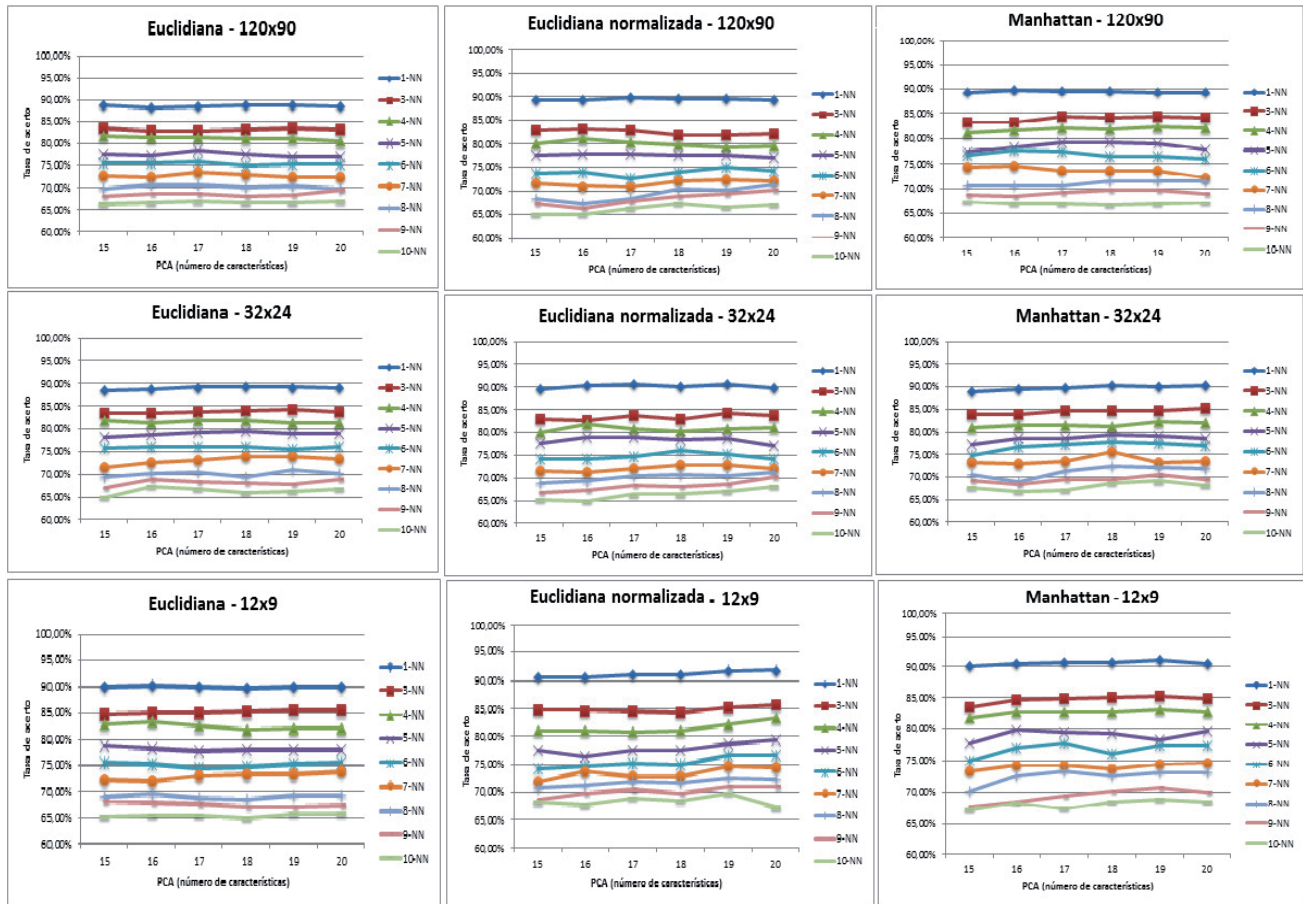


Figura 5. Resultados da Taxa de classificação para as imagens com dimensões 120x90, 32x24 e 12x9.

Os algoritmos *Random Forest* e *K-star* foram simulados utilizando o Software WEKA. Os experimentos seguiram a metodologia *10-fold-cross-validation* [Ghosal et al, 2009]. As sequências foram aleatoriamente separadas em 10 grupos de tamanhos aproximadamente iguais. Um grupo foi utilizado como conjunto de novas sequências apresentadas ao sistema para o aprendizado e os 9 grupos restantes foram utilizados como sequências de treinamento dos algoritmos. No *Random Forest* varia-se o número de árvores e o número de características a serem analisadas. As variações testadas e os resultados se encontram na Tabela 1. Nesta tabela, pode-se observar que a melhor taxa de acerto resultante foi 91,4%, utilizando 450 árvores e 2 características.

Tabela 1. Testes *Random Forest*.

Número de árvores	Número de Características	Taxa de acerto
10	5	78.2292%
50	5	87.5%
100	3	89.6875%
100	5	88.5417%
200	3	90.9375%
250	3	90.4167%
300	3	90.625%
400	3	90.8333%
450	2	91.4583%
450	3	91.0417%

No *K-Star*, os valores testados para o *global blend* (descrito na subseção 3.3), assim como os resultados

obtidos estão descritos na Tabela 2. A melhor taxa de acerto resultante foi aproximadamente 91%, utilizando um *Global Blend* de 100%.

Tabela 2. Testes *K-Star*.

<i>Global Blend</i>	Taxa de acerto
20%	88.4375%
50%	89.375%
100%	91.0417%

Analisando os 3 métodos, ambos tiveram como melhores resultados valores parecidos, em torno de 90% acima. Destacando o melhor resultado obtido de testes com a técnica *K-NN* em imagens de dimensões 12x9 e utilizando números de 20 características resultantes do PCA, atingindo uma taxa de reconhecimento de aproximadamente 92%.

Analisando também os métodos *Random Forest* e *K-Star*, em relação ao requisito do tempo, observou-se que o algoritmo *Random Forest* consome maior tempo de execução na etapa de treinamento do modelo do que na etapa de teste do modelo treinado. Já o algoritmo *K-Star* possui comportamento diferente, consumindo a maior parte do tempo de execução na etapa de teste do modelo treinado. Logo, conseqüentemente esses algoritmos têm um desempenho menor que o algoritmo KNN na classificação das faces.

6. Discussões dos Experimentos

Na Figura 3, os gráficos para imagens de face de dimensão 120 x 90 verifica-se a maior taxa de acerto

(89,9%), usando o $k=1$, número de autovetores igual a 17 e distância euclidiana normalizada. A menor taxa de acerto encontra-se quando usa o $k=10$ e distância euclidiana normalizada. Vale destacar aqui que para os outros valores de k os melhores resultados se encontram utilizando a distância Manhattan. Nos gráficos das faces de dimensão 32×24 (Figura 3) mostra que a maior taxa de acerto (90,63%) é usando o $k=1$, número de autovetores igual a 19 e distância euclidiana normalizada. A menor taxa de acerto encontra-se quando usa o $k=10$ e distância euclidiana. Vale destacar aqui também que para os outros valores de k os melhores resultados se encontram utilizando a distância Manhattan. Além disso, há uma uniformidade dos melhores resultados, se encontrando entre os valores característicos do PCA variando entre 18 e 20.

Já nos gráficos das faces de dimensão 12×9 (Figura 3) verifica-se que a maior taxa de acerto (91,98%) foi usando o $k=1$, número de autovetores igual a 20 e distância euclidiana normalizada. A menor taxa de acerto encontra-se quando se usa o $k=10$ e distância euclidiana. Vale destacar aqui que para os outros valores de k os melhores resultados se encontram utilizando a distância euclidiana normalizada, com os melhores resultados se encontrando entre os valores característicos do PCA variando entre 19 e 20. Concluímos com os resultados obtidos nas três dimensões (Figura 3) que se os valores de k crescem, implica na diminuição da precisão da classificação. Além disso, quanto menor a dimensão do pixel, maior é a resolução espacial da imagem. Quanto maior a resolução espacial da imagem, maior serão a definição e detalhe da imagem. Uma imagem com resolução baixa ficará borrada e com menos detalhes, diminuindo o reconhecimento de padrão. Isto favoreceu os melhores resultados encontrados nas imagens de dimensão 12×9 .

Já analisando os 3 algoritmos, ambos tiveram como melhores resultados valores parecidos, em torno de 90% acima. Destacando o melhor resultado na taxa de acerto obtido de testes com a técnica K-NN em imagens de dimensões 12×9 e utilizando números de 20 características resultantes do PCA, atingindo uma taxa de reconhecimento de aproximadamente 92%. Além disso, a técnica KNN tem um desempenho melhor que o *Random Forest* e *K-Star*, apresentando menores tempos na classificação das faces.

7. Conclusões e trabalhos futuros

Neste trabalho, são tratados problemas de dimensionalidade para o reconhecimento facial. A abordagem utilizando *Eigenfaces* e PCA é bastante robusta no tratamento de diferentes imagens faciais, em suas diversas direções. O *Eigenface* provou ter capacidade de fornecer as características significativas e reduz o tamanho de entrada para os algoritmos de classificação, melhorando o reconhecimento.

No entanto, esta abordagem mostrou-se ser sensível em imagens com condições de iluminação não controladas. Experimentalmente, verificamos que a abordagem de reconhecimento facial utilizando PCA, *Eigenface* e as técnicas de classificação K-NN, *Random Forest* e *K-Star*

trouxeram bons resultados. Todos esses três classificadores atingiram uma acurácia máxima de 90%. O melhor resultado foi obtido de testes com a técnica K-NN em imagens de dimensões 12×9 e utilizando números de 20 características resultantes do PCA, atingindo uma taxa de reconhecimento de aproximadamente 92%.

Para trabalhos futuros, utilizando a abordagem de PCA e *Eigenface*, será feita uma análise dos algoritmos de aprendizado supervisionado, tais como: Redes Neurais [Agarwal *et al.*, 2010]. Devido aos bons resultados obtidos neste trabalho, também será feito, em paralelo, um sistema de reconhecimento de expressões faciais utilizando a abordagem de PCA e *Eigenface* e o algoritmo K-NN como classificador das emoções faciais.

Referências bibliográficas

- [Agarwal *et al.*, 2010] Agarwal, M.; Agrawal, H.; Jain, N. Kumar, M. "Face Recognition Using Principle Component Analysis, Eigenface and Neural Network", Signal Acquisition and Processing. ICSAP 10. International Conference on, vol., no., pp. 310-314, 9-10, 2010.
- [Breiman, 2001] Breiman, L. Random Forests. In Journal of Machine Learning, Vol.45, pages 5-32. Kluwer Academic, Netherland, 2001.
- [Cleary;Trigg, 1995] CLEARY, J.G.; TRIGG, L.E. K*: an instance-based learner using an entropic distance measure. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 12., 1995, Tahoe City, California, US . Proceeding. Tahoe City, 1995. p.108-114.
- [Ekman, 1993] Ekman, P. "Facial Expression and Emotion. American Psychologist", v.48, n.4, p. 384-392, 1993.
- [Frank *et al.*, 1999] Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann P.; Witten, I. H. The WEKA data mining software: an update. SIGKDD Explorations, v. 11, n. 1, University Mark Hall, 1999.
- [Ghosal *et al.*, 2009] Ghosal V.; Tikmani P.; Gupta P. "Face Classification Using Gabor Wavelets and Random Forest", In Proceedings of the Canadian Conference on Computer and Robot Vision (CRV '09), IEEE Computer Society, Washington, DC, USA, 68-73, 2009.
- [Hyun-Chul; Se-Young, 2012] CHOI, Hyun-Chul; OH, Se-Young. Multi-Face Detection on static image using Principal Component Analysis. ICCAS 2004. Disponível em: < <http://isl.postech.ac.kr> >. Acesso em: Março 2012.
- [Jiangsheng, 2002] Jiangsheng, Y. Method of k-Nearest Neighbors. Institute of Computational Linguistics, Peking University, China, 2002.
- [Kshirsagar *et al.*, 2011] Kshirsagar, V. P.; Baviskar, M. R.; Gaikwad, M. E. "Face recognition using Eigenfaces", Computer Research and Development (ICCRD), 3rd International Conference on, vol. 2, no., pp. 302-306, 11-13, 2011.
- [Romdhani, 1996] Romdhani, S. Face Recognition Using Principal Component Analysis. University of Glasgow (Dissertação). 1996.

- [Segal, 2004] Segal, M. R. Machine learning benchmarks and random forest regression. http://www.ics.uci.edu/_mlearn/MLRepository.html. Center for Bioinformatics & Molecular Biostatistics, 2004.
- [Stan;Anil, 2011] Stan Z. Li; Anil K. J. “Handbook of Face Recognition, 2nd Edition”, Springer, 2011.
- [Turk, 1991] Turk M. and Pentland A. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71-86, 1991.
- [Weka, 2011] Weka (software) “Data Mining Software in Java The University of Waikato”. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>, Acesso em: 05 de nov. de 2011.
- [Zainal; Maarof, 2008] Zainal, A.; Maarof, M.A., Shamsuddin, S.M. and Abraham, A. Ensemble of One-class Classifiers for Network Intrusion Detection System. In Proceedings of Fourth International Conference on Information Assurance and Security, pages 180-185, 2008.
- [Zhang; Ruan, 2006] Zhang B.; Ruan Q. “Facial feature extraction using improved deformable templates,” Signal Processing, 8th International Conference on, vol.4, no., 16-20, 2006.