RedFace: um sistema de reconhecimento facial baseado em técnicas de análise de componentes principais e autofaces: comparação com diferentes classificadores

Fábio Abrantes Diniz¹
Francisco Milton Mendes Neto¹
Francisco das Chagas Lima Júnior¹
Laysa Mabel de O. Fontes¹

Resumo: Desenvolver um modelo computacional de reconhecimento facial não é uma tarefa trivial, eis que 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 essas 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 foram utilizadas 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, ficou constatado que o desempenho dos algoritmos testados para sistemas de reconhecimentos de face baseado em PCA foram satisfatórios, atingindo as melhores taxas de reconhecimento, acima de 90% em todos os classificadores. Ao descobrir a melhor técnica, este será aplicado na identicação de alunos que acessam um ambiente virtual de aprendizagem.

Palavras-chave: Análise de componentes principais. Autoface. Reconhecimento facial.

Abstract: Developing a computational model for facial recognition is not an easy task, because faces and multi-dimensional visual stimulation have complex modeling features. The difficulty lies in a face modeling which set aside the features that differ from other faces, since they have a few substantial differences among themselves. Although different, all faces have features such as: a mouth, two eyes and a nose. In this work, we propose a facial recognition system developed in two stages. First, were used techniques of Principal Components Analysis (PCA) and Eigenfaces in order to extract the face features. Then, the classifiers K-Nearest Neighbors (K-NN), Random Forest and K-Star will be applied in the face recognition process. The algorithms validation was held in a database with 1280 images of 64 different classes. Finally, was evidenced that the performance tests of the algorithms in face recognition systems based on PCA were consider very satisfactory, reaching the best recognition rates above 90% in all classifiers. Finding out the best technique, it will be applied in the students identification who access a Virtual Learning Environment.

Keywords: Eigenfaces. Face Recognition. Principal component analysis.

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

http://dx.doi.org/10.5335/rbca.2013.2627

¹ Programa de Pós-Graduação em Ciência da Computação, UERN/UFERSA, BR 110 - Km 46 - Bairro Costa e Silva - Campus Central, 59.625-620, Mossoró - RN, Brasil.

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 [5]. 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 [1].

Existe uma variedade de métodos para a identificação de usuários, os quais são utilizados no controle de acesso em aplicações comerciais [11]. Porém, nos ambientes virtuais de aprendizagem (AVAs), essas técnicas de identificação não estão sendo abordadas, pois as restrições de ordem econômica e operacional são diferentes [11]. Em geral, alguns desses ambientes adotam a utilização de mecanismos de identificação por *login* e senha como prática de segurança. O uso desse tipo simples de autenticação aumenta a vulnerabilidade a fraudes, tanto no acesso ao sistema quanto durante a participação do aluno nas atividades do curso, pois outra pessoa pode substituir facilmente o indivíduo que deveria ser avaliado no AVA acessado [13].

Baseado nisso, este trabalho propõe analisar a acurácia de algumas técnicas no reconhecimento de faces. A técnica que obtiver o melhor resultado será aplicado a um AVA, adicionando a funcionalidade de autenticação biométrica. Vai monitorar um estudante, capturando imagens através de uma câmera, bem como detectará a face do estudante na imagem e identificará dentre outras faces cadastradas no sistema.

Diferentes técnicas foram utilizadas, neste trabalho, para a construção de um sistema de reconhecimento facial. 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) [16] no processo de extração de características da face para diminuir a alta dimensionalidade dos vetores [16]. A classificação das imagens foi realizada através de análises dos classificadores *K-Nearest Neighbors* (K-NN) [9], *Random Forest* (Floresta aleatória) [2] e *K-Star* (K-estrela) [4]. O propósito foi descobrir qual combinação dessas técnicas tem melhor desempenho em termos de taxa de reconhecimento facial.

O presente trabalho está organizado como segue. A 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 [16]. A primeira é baseada na extração de vetores característicos de partes básicas de uma face, tais como: olhos, nariz, boca e queixo. Essa 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 [16]. A segunda abordagem é baseada nos conceitos da teoria da informação [16]. Nessa 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 [21].

Em Ghosal, Tikmani e Gupta [7], 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*, um 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 *Ramdon 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, Baviskar e Gaikwad [10], os métodos de análise dos componentes principais (PCAs) e os métodos estatísticos foram utilizados. Em Stan e Anil [16], 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, essa 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 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 Sistema 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 [16]. A arquitetura do sistema de reconhecimento facial é apresentada na Figura 1.

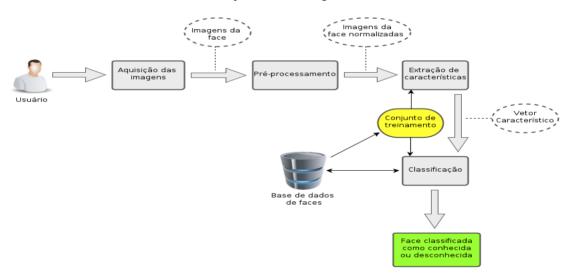


Figura 1. Arquitetura do sistema de reconhecimento facial

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. Nesse módulo foi utilizado o algoritmo para detecção de face *Viola-Jones* [18]. Esse algoritmo tenta encontrar em uma imagem características que codificam alguma informação da classe a ser detectada. Para tal tarefa, são usadas as chamadas características de *Haar*, responsáveis por codificar informações sobre a existência de contrastes orientados entre regiões da imagem. As imagens foram amostradas quadro a quadro e as faces foram detectadas e extraídas usando o algoritmo Viola-Jones na implementação da biblioteca OpenCV [12].

OpenCV é uma biblioteca, totalmente livre para o uso acadêmico e comercial, desenvolvida pela Intel no ano de 2000 voltada para o desenvolvimento de aplicações de tempo real no campo da visão computacional. Originalmente desenvolvida em C/C++, ela é portável para todas as plataformas de sistemas operacionais. O OpenCV possui módulos de processamento de imagens e vídeo, estrutura de dados, álgebra linear, interface gráfica do usuário (GUI), controle de mouse e teclado, além de mais de mais de 2500 algoritmos, muitos dos quais são considerados estado da arte, tais como segmentação, reconhecimento de faces (método Viola-Jones), aprendizado de máquinas, filtragem de imagens, rastreamento de movimento, entre outros métodos [12].

O próximo módulo é o pré-processamento da imagem. Nesse módulo, as imagens da face detectadas pelo algoritmo Viola-Jones são automaticamente normalizadas e corrigidas para melhorar o reconhecimento da face. As imagens tiveram correções nas suas dimensões (todas foram redimensionadas somente para as dimensões 120x90, 32x24 e 12x9), na resolução (300 dpi) e na iluminação (foi utilizada a mesma em todas as fotos) e foram transformadas para escala de cinza. Esse módulo é útil para poder fazer as comparações com as imagens de faces cadastradas no banco de dados a qual é formada somente por imagens das dimensões 120x90, 32x24 e 12x9.

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 x h, onde w e h são, respectivamente, os números de pixels referentes à largura e à altura da imagem [12]. Cada valor do pixel da imagem corresponde a uma componente do vetor.

Devido à alta dimensionalidade dos vetores, foi utilizada, nessa etapa, a técnica de PCA a qual reduz a quantidade de características de uma imagem. Logo, consequentemente, diminui o custo computacional e melhora a precisão do classificador. Se o espaço característico contiver somente as características mais visíveis, o classificador será mais rápido e ocupará menos memória [1]. 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 [9], Random Forest [2] e K-Star [4] os quais são detalhados nas subseções seguintes. Esses dois últimos foram utilizados a partir do software de mineração de dados Weka (Waikato Environment for Knowledge Analysis) [19].

O Weka é um software de mineração de dados, desenvolvido na linguagem Java, muito popular no meio acadêmico. É formado por um conjunto de algoritmos de diversas técnicas de Mineração de Dados [6], [19]. O Weka tem como principal característica a sua portabilidade, dessa forma, é possível utilizá-la em diferentes sistemas operacionais, além de aproveitar os principais beneficios da orientação a objetos. O Weka é um software livre e está no domínio da licença GPL (General Public License) [6], [19].

Alguns métodos implementados no Weka [6], [19]:

- Métodos de classificação: árvore de decisão induzida, regras de aprendizagem, tabelas de decisão, regressão local de pesos, aprendizado baseado em instância, regressão lógica, perceptron, perceptron multicamada, comitê de perceptron, Random Forest, entre outros;
- **Métodos para predição numérica:** regressão linear, geradores de árvores modelo, regressão local de pesos, aprendizado baseado em instância, tabela de decisão e entre outros;
- **Métodos de agrupamento:** *Cobweb*, *SimpleKMeans*, *DBScan*, *CLOPE*, métodos de associação, *Apriori*, *FPGrowth*, *PredictiveApriori*, *Tertius*; *K-Star* entre outros

O último módulo da Figura 1 é o conjunto de treinamento, que 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.

3.1 Classificador K-Nearest Neighbors

O K-NN é um dos algoritmos de classificação mais utilizados na área de aprendizagem de máquina [9]. É baseado na procura dos k vizinhos mais próximos do padrão de teste. 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 [9]. 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 [9]. 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 Forest* foi proposto em Breiman [2]. Ele consiste em uma técnica de agregação de classificadores do tipo árvore, construídos de forma que a sua estrutura seja composta de maneira aleatória [7]. 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 [2].

O classificador é baseado no método *Bagging* [2]. 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 [2].

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 um superajuste (*overfitting*) em determinadas bases [15].

Como mencionado anteriormente, esse algoritmo foi utilizado a partir da ferramenta Weka [19], 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 [20].

3.3 Classificador K-Star

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 [4]. 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 [6]. O *K-Star* possui os parâmetros de configuração para a execução os quais são *globalBlend* e *missingMode* [19].

Para a execução do K-Star, é necessário configurar dois parâmetros [6], [19], abaixo descritos:

- *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 [10].

O PCA é baseado nos autovetores e autovalores da matriz covariância dos dados [1]. 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:

$$C = XXT = V\Lambda V^T \in R^{nxn} \tag{1}$$

Em que $\Lambda = diag \{\lambda_1,...,\lambda_n\}$ é a matriz diagonal contendo n autovalores, v_i é o autovetor associado a λ_i e $V = [v_1,...,v_n] \in R_{nxn}$ é a matriz dos autovetores, v_i , da matriz covariância

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \cdots & \mathbf{X}_{1m} \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{n1} & \cdots & \mathbf{X}_{nm} \end{bmatrix}$$
(2)

onde X é a matriz de dados originais, em que $x_{ij} \in R$, i = 1,..., n e j = 1,..., m. O PCA determina uma transformação linear dos elementos de X, tal que:

$$y = V^T X \tag{3}$$

onde $y = [y_1, y_2,..., y_n]$ 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 [10]. 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 [17].

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. Esse 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 [17].

A sequência dos procedimentos da geração das Eigenfaces abordados por esse trabalho foi [3]:

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

$$\Gamma = [\Gamma_1, \ \Gamma_2, \dots \Gamma_M] \tag{4}$$

II. Calcular a face média $\Psi(5)$, para eliminar muita informação redundante na face, isto é, uma imagem de face poderia ser representada com poucas componentes principais devido à sua redundância.

$$\Psi = \frac{\sum_{i=1}^{M} r_i}{M}$$
 (5)

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 (6) que contém somente as variações de cada face em relação à face média. O vetor Φ (7) contém todas as variações de uma determinada face Γ em relação à face média Ψ .

$$A = [\Phi 1, \Phi 2, \dots, \Phi m] \tag{6}$$

$$\Phi = \Gamma - \Psi \tag{7}$$

V. Sendo M menor que a dimensionalidade (largura multiplicado pela altura das imagens de treinamento) das imagens em Φ , calcular a matriz de covariância C(8).

$$C = AA^{T} (8)$$

VI. Calcular os autovetores ν e autovalores λ da matriz C.

VII. Criar a matriz de transformação. A matriz μ (9) conterá M - I autovetores significativos. Entretanto, ainda é possível realizar a eliminação de alguns desses M-I autovetores pela ordem de importância de seus correspondentes autovalores λ , gerando M autovetores escolhidos.

$$\mu = \nu \Phi \tag{9}$$

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 (9). Em seguida o novo padrão passa para a fase de classificação.

$$\Omega = \mu^T \Phi \tag{10}$$

A Figura 4 apresenta a face média do conjunto de treinamento (Fig. 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 a 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 [8] e 12x9 – melhor resultado dos testes automáticos. Além disso, todas as imagens faciais tiveram correções nas suas iluminações, ou seja, todas as imagens tiveram as mesmas iluminações, pois, com imagens pouco iluminadas, o método *Eigenface* perde sua eficiência na extração das características.

Nas simulações deste trabalho, foi utilizado um total de 64 pessoas, com dados extraídos de uma base de dados própria, possuindo 1280 imagens de um total de 64 pessoas. Essas imagens são de pessoas conhecidas dos autores deste trabalho. Essas pessoas estão cadastradas num ambiente virtual de aprendizagem, ao qual esse

sistema de reconhecimento facial será integrado, logo, constituindo-se como um principal motivo para a utilização dessa base de dados própria para os testes dos métodos abordados neste trabalho. No entanto, em trabalhos futuros serão utilizadas bases de imagens padronizadas [22], [23], [24], [25] para comparações de resultados, verificando a taxa de acerto em relação à literatura existente. 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 vinte imagens nas seguintes posições: quatro imagens olhando diretamente para a câmera, quatro olhando 45 graus à esquerda da câmera, quatro olhando 45 graus à direita da câmera, quatro olhando 90 graus à esquerda da câmera e quatro imagens olhando 90 graus à direita da câmera, como pode ser visto na Figura 3. Dessas vinte imagens, cinco (cada uma de uma posição) compõem o conjunto de treinamento do sistema e as 15 restantes são destinadas para o conjunto de teste.



Figura 3. Esquema de posicionamento da face.

Em seguida, os testes foram desenvolvidos tentando encontrar o melhor número de características restantes após o PCA (número de autovetores do *Eigenfaces*). Foram utilizados testes com valores de características no intervalo de 15 a 120 para encontrar os melhores resultados tanto no reconhecimento da face

como no custo computacional. 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: 17 características para 120x90; 19 características para 32x24 e 20 características para 12x9.

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 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.

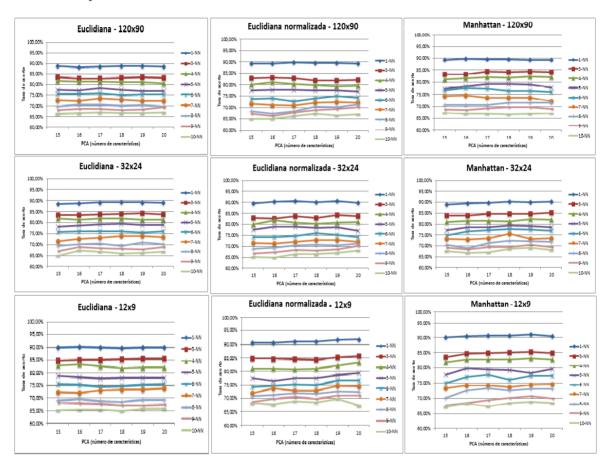


Figura 5. Resultados da taxa de classificação para as imagens com dimensões 120x90, 32x24 e 12x9.

Os algoritmos *Random Forest* e *K-star* foram simulados utilizando o Software Weka [19]. Os experimentos seguiram a metodologia 10-fold-cross-validation [7]. As sequências foram aleatoriamente separadas em dez grupos de tamanhos aproximadamente iguais. Um grupo foi utilizado como conjunto de novas sequências apresentadas ao sistema para o aprendizado e os nove 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. Verifica-se, nessa Tabela, que aumentar o número de características e manter o número de árvores implica a diminuição da taxa de acertos. Isso pode ser visto nas linhas três e quatro da Tabela 1, com ambas mantendo número de árvore de cem, e também nas linhas nove e dez da Tabela 1, com ambas mantendo o número de árvore de 450. Nessa tabela, pode-se observar que a melhor taxa de acerto resultante foi 91,4%, utilizando 450 árvores e duas 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 acero resultante foi aproximadamente 91%, utilizando um *Global Blend* de 100%.

Tabela 2: Testes K-Star.

Global Blend	Taxa de acerto
20%	88,4375%
50%	89,375%
100%	91,0417%

6 Discussões dos experimentos

Na Figura 5, no que concerne aos gráficos para imagens de face de dimensão 120×90 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 Euclidiana normalizada. Nos gráficos das faces de dimensão 32×24 (Fig. 5) é possível perceber que a maior taxa de acerto (90,63%) se dá quando usado 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 Euclidiana. Além disso, há uma uniformidade dos melhores resultados, se encontrando entre os valores característicos do PCA variando entre 18 e 20.

Nos gráficos das faces de dimensão 12x9 (Fig. 5) verifica-se que a maior taxa de acerto (91,98%) foi usando o k=1, número de autovetores igual a vinte 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 (Fig. 5), que o crescimento dos valores de k implica a 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 o detalhe da imagem. Uma imagem com resolução alta ficará menos borrada e com mais detalhes, melhorando o reconhecimento do padrão. Isso favoreceu os melhores resultados encontrados nas imagens de dimensão 12x9.

Analisando os três algoritmos, todos 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 12x9 e utilizando números de vinte características resultantes do PCA, atingindo uma taxa de reconhecimento de aproximadamente 92%. Além disso, a técnica K-NN tem um desempenho melhor que o *Random Forest* e *K-Star*, apresentando menores tempos na classificação das faces.

Analisando também os métodos *Random Forest*, *K-Star* e K-NN, em relação ao requisito do tempo, somente será feito, em trabalhos futuros, com os métodos *Eigenfaces* e o classificador K-NN, durante o momento que esse sistema de reconhecimento facial for integrado ao Ambiente Virtual de Aprendizado (AVA), pois o requisito de tempo neste AVA é obrigatório. Além disso, os algoritmos *Random Forest* e *K-Star* são soluções com uso de terceiros, logo uma análise de tempo para esses algoritmos não oferece uma visão detalhada e confiável.

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, essa abordagem mostrou ser sensível em imagens com condições de iluminação não controladas, logo, em um trabalho futuro, serão utilizadas bases de dados de imagens faciais consolidadas na literatura. Experimentalmente, utilizando os recursos deste trabalho, verificou-se 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 conseguiram atingir acurácias acima de 90% em específicos parâmetros. O melhor resultado foi obtido de testes com a técnica K-NN em imagens de dimensões 12x9 e utilizando números de vinte características resultantes do PCA, atingindo uma taxa de reconhecimento de aproximadamente 92%. Logo, o classificador K-NN, juntamente com as técnicas *Eigenface* e PCA será utilizado numa aplicação de ensino a distância para identificação de alunos durante todo seu acesso no curso não presencial.

Dessa forma, em trabalhos futuros, utilizando a abordagem de PCA e Eigenface, será feita uma análise dos algoritmos de aprendizado supervisionado, tais como: Redes Neurais, Máquinas de Vetor de Suporte e Rede *Bayesianas*. Além disso, serão feitos testes com outras bases de dados consolidadas nas bibliografías existentes a fim de verificar sua acurácia com os outros resultados de trabalhos propostos para o reconhecimento facial. Devido aos bons resultados obtidos neste trabalho utilizando a base de dados criada, 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.

Agradecimentos

Os autores agradecem à CAPES pela concessão das bolsas de pesquisa e pelo apoio financeiro para a realização desses estudos.

Referências

- [1] 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.
- [2] Breiman, L.; Random Forest. In Journal of Machine Learning, Vol.45, pages 5-32. Kluwer Academic, Netherland, 2001.
- [3] Cheon, Y.; Kim, D. "A Natural Facial Expression Recognition Using Differential-AAM and k-NNS," Multimedia, Tenth IEEE International Symposium on, vol., no., pp.220-227, 15-17, 2008.
- [4] 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.
- [5] Ekman, P. "Facial Expression and Emotion. American Psychologist", v.48, n.4, p. 384-392, 1993.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Jiangsheng, Y. Method of k-Nearest Neighbors. Institute of Computational Linguistics, Peking University, China, 2002.
- [10] 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.
- [11] MARAIS, E.; Argles D.; von Solms B. Security issues specific to e-assessments, 8th Annual Conference on WWW Applications, 2006.
- [12] OPENCV Open Source Computer Vision Library: biblioteca de visão computacional open source. Estados Unidos, Intel Corporation, 2001 [436 p.]. Disponível em: http://developer.intel.com Acesso em: 12/01/2012.
- [13]RABUZIN, K.; Baca, M.; Sjako, M. E-learning: biometrics as a security factor, Proceedings of the International Multi-Conference on Computing in the Global Information Technology, pp. 64-74, 2006
- [14] Romdhani, S. Face Recognition Using Principal Component Analysis. University of Glasgow (Dissertação). 1996.
- [15] Segal, M. R. Machine learning benchmarks and random forest regression (http://www.ics.uci.edu/_mlearn/MLRepository.html). Center for Bioinformatics & Molecular Biostatistics, 2004.
- [16] Stan Z. Li; Anil K. J. "Handbook of Face Recognition, 2nd Edition", Springer, 2011.
- [17] Turk M. and Pentland A. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71 [86, 1991.
- [18] Viola, P. A.; Jones, M. J. "Robust real-time object detection", International Journal of Computer Vision, 57(2): 137-154, 2004.
- [19] Weka (software) "Data Mning Software in Java The University of Waikato". Disponível em: http://www.cs.waikato.ac.nz/ml/weka/, Acesso em: 05 de nov. de 2011.
- [20] 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.

- [21] Zhang B.; Ruan Q. "Facial feature extraction using improved deformable templates," Signal Processing, 8th International Conference on, vol.4, no., 16-20, 2006.
- [22] Gross, R.; Matthews, I.; Cohn, J.; Kanade, T.; Baker, S.; "Multi-PIE Automatic Face & Gesture Recognition," FG '08. 8th IEEE International Conference on , vol., no., pp.1-8, 17-19, 2008.
- [23]Mislav,G.; Kresimir. D.; Sonja G. "SCface --- surveillance cameras face database". Multimedia Tools Appl. 51, 3 (February 2011), 863-879, 2011.
- [24] Loui, A.C.; Judice, C.N.; Sheng Liu; "An image database for benchmarking of automatic face detection and recognition algorithms," Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, vol.1, no., pp.146-150 vol.1, 4-7, 1998.
- [25]Sharma, P.; Reilly, R.B.; "A colour face image database for benchmarking of automatic face detection algorithms," Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on , vol.1, no., pp. 423- 428 vol.1, 2-5, 2003.