

# A comparative study involving classifiers and dimensionality reduction techniques applied to facial recognition

Matheus S. Araújo<sup>1</sup>, Leonardo F. da Costa<sup>1</sup>, Antony K. F. Santos<sup>1</sup>,  
Caio G. de O. Menezes<sup>1</sup>, Gustavo A. L. de Campos<sup>1</sup>

<sup>1</sup>Universidade Estadual do Ceará (UECE)  
Caixa Postal (85) 3101-9601 – 60.714.903 – Fortaleza – CE – Brazil  
Curso de Ciência da Computação – Campus do Itaperi

{math.araujo, leonardo.costa}@aluno.uece.br

{kevin.antony.santos, cgomenezes}@gmail.com

gustavo@larces.uece.br

**Abstract.** *This paper presents a comparative study between the Eigenfaces and Fisherfaces techniques combined with the KNN, SVM and MLP classifiers. The Eigenfaces and Fisherfaces techniques were used to project the images from the AT&T (The database of faces) and Extended Yale B databases into a new space in order to obtain a reduction in the dimensionality of these data. The classifiers mentioned used the data designed to perform the training task and subsequent identification of the test data classes. The results were very promising in both cases, but the MLP neural network with the Fisherfaces technique obtained the best results.*

**Resumo.** *Neste trabalho é apresentado um estudo comparativo entre as técnicas Eigenfaces e Fisherfaces combinadas com os classificadores KNN, SVM e MLP. As técnicas Eigenfaces e Fisherfaces foram utilizadas para projeção das imagens dos bancos de imagens AT&T (The database of faces) e Extended Yale B em um novo espaço de forma a se obter uma redução da dimensionalidade desses dados. Os classificadores mencionados utilizaram os dados projetados para executar a tarefa de treinamento e posterior identificação das classes dos dados de teste. Os resultados foram bastante promissores em ambos os casos, porém a rede neural MLP com a técnica Fisherfaces obtiveram os melhores resultados.*

## 1. Introdução

Apesar de o problema de se reconhecer faces ser trivial para seres humanos, para computadores, este problema mostrou-se quase impossível de se resolver [Coppin 2010]. O reconhecimento facial é considerado uma das técnicas biométricas mais importantes e possui vantagens em relação às demais técnicas biométricas, como o fato de ser um processo natural e passivo, não necessitando da cooperação de indivíduos como acontece em outras técnicas, como o reconhecimento de impressões digitais [Zhang and Gao 2009].

Embora o reconhecimento de impressões digitais seja mais preciso que outros métodos de reconhecimento, o reconhecimento facial ainda atrai grande atenção de muitos

pesquisadores no campo de visão computacional. Existem duas categorias de algoritmos utilizados em reconhecimento de faces: os métodos baseados em características e métodos baseados em aparência [Silva 2009, Eleyan and Demirel 2007].

Os métodos baseados em características utilizam características faciais distintas, como olhos, boca e nariz para obter o vetor de características, já os métodos baseados em aparência extraem características globais da imagem, observando os valores dos píxeis para isso [Silva and Cintra 2015]. Essa extração de características ou extração de atributos, consiste em obter através dos dados originais, novos dados que melhor descrevem as entidades em estudo [Müller and Guido 2016]. Em geral, métodos baseados em aparência são mais robustos que métodos baseados em características [Tan et al. 2002].

Esses métodos geralmente implicam na representação de imagens como um vetor, que na prática é muito grande para permitir o reconhecimento robusto e rápido de objetos [Martínez and Kak 2001]. Uma forma muito comum de se resolver tal problema é através do uso de técnicas de redução de dimensionalidade, como por exemplo, as técnicas PCA e LDA.

A análise de componentes principais (*Principal Component Analysis* ou PCA) é um método usado tanto para redução de dimensionalidade em problemas de compressão, quanto para problemas de reconhecimento [Toygar and Adnan 2003]. A aplicação desse método no contexto de reconhecimento facial é também referida como Eigenfaces [Turk and Pentland 1991a].

A análise de discriminantes lineares (*Linear Discriminant Analysis* ou LDA) é uma poderosa técnica de redução de dimensionalidade que procura tornar os dados de classes diferentes os mais separados possível, para isso essa técnica utiliza uma análise da matriz de dispersão dos dados [Toygar and Adnan 2003]. A aplicação dessa técnica em problemas de reconhecimento facial é também conhecida como Fisherfaces [Belhumeur et al. 1997].

Este trabalho então tem como objetivo desenvolver um estudo comparativo quanto à eficiência das técnicas de extração de atributos Eigenfaces (PCA) e Fisherfaces (LDA) no problema de reconhecimento facial, empregando classificadores clássicos de aprendizagem de máquina: o simples e conhecido algoritmo KNN (*K-nearest neighbors*); a rede neural Perceptron multicamadas (*Multilayer Perceptron* o MLP); e um modelo de máquina de vetores de suporte (*Support Vector Machine* ou SVM).

## 2. Metodologia

A Figura 1 apresenta uma representação esquemática do estudo realizado neste trabalho.

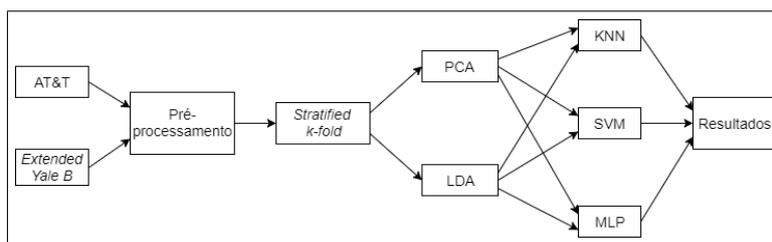


Figura 1. Esquema geral do estudo.

As bases de dados utilizadas nos experimentos foram as bases de dados públicas de faces AT&T (*The database of faces*) e *Extended Yale database B*.

A base de dados AT&T [AT&T Laboratories Cambridge 2002] possui imagens das faces de 40 indivíduos (36 homens e 4 mulheres), com 10 imagens distintas para cada indivíduo, totalizando 400 imagens. Existem dez imagens diferentes de cada uma das 40 pessoas do banco de imagens. As imagens foram tiradas em diferentes momentos, variando a iluminação, expressões faciais (olhos abertos / fechados, sorrindo / sem sorrir) e detalhes faciais (óculos / sem óculos). Todas as imagens foram tiradas contra um fundo escuro homogêneo com os sujeitos em posição vertical e frontal (com tolerância para algum movimento lateral).

A base de dados *Extended Yale Database B* [Georghiadis et al. 2001] contém imagens de 38 pessoas sob 64 condições de iluminação. Neste trabalho foram utilizadas as imagens de face recortadas, que consistem de 2414 imagens frontais da face dos 38 indivíduos [Lee et al. 2005], possuindo dimensões 192 x 168 e sendo capturadas em vários condições de iluminação controladas por laboratório.

As imagens da base de dados AT&T foram usadas sem qualquer pré-processamento, ou seja, houve apenas a extração dos valores dos píxeis das imagens de face. Já as imagens da base de dados *Extended Yale B* foram redimensionadas de 192 x 168 para 100 x 88 para reduzir a complexidade.

Para uma melhor confiabilidade nos resultados, foi utilizado o método *Stratified k-fold*, com  $k = 10$ , como estratégia de divisão da base de dados. Dessa forma, a base de dados AT&T foi dividida em dez subconjuntos, usando uma imagem de face de cada indivíduo em cada subconjunto. Logo, cada subconjunto criado possui exatamente 40 imagens, cada uma correspondendo a um indivíduo.

Em cada uma das dez iterações do método *Stratified k-fold*, nove subconjuntos são utilizados para treinamento, totalizando 360 imagens para treinamento, e um subconjunto é utilizado para testes, totalizando 40 imagens para teste. De forma análoga, para a base de dados *Extended Yale B*, que possui 2414 imagens de face de 38 indivíduos, o número de imagens em cada *fold* ou subconjunto é de aproximadamente 241 imagens, de forma que em cada *fold* pode-se ter em média 6 imagens para cada indivíduo.

As próximas subseções descrevem os procedimentos para Eigenfaces e Fisherfaces de extração de características.

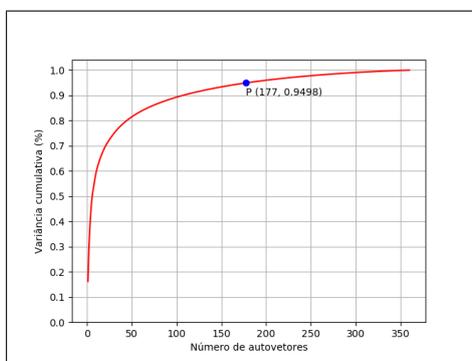
## 2.1. Eigenfaces - PCA

Foi criado um espaço de faces do método Eigenfaces a partir das faces que foram selecionadas para treinamento. Cada face do conjunto de treinamento será projetada no espaço de faces *eigenspace*, de forma que cada imagem será representada por um vetor de pesos que será utilizado para treinar os classificadores KNN, SVM e MLP. Note que esse processo de divisão e criação de um espaço de faces será feito dez vezes, devido ao uso do método *Stratified k-fold*.

Segundo [Turk and Pentland 1991b] o número total de *eigenfaces* é igual ao número de faces utilizadas para criar o espaço *eigenface space*, portanto, para a base de dados AT&T o número total de *eigenfaces* é 360, pois em cada iteração do método *Stratified k-fold*, 9 *fold's*, de 40 imagens cada, foram utilizados para treinamento.

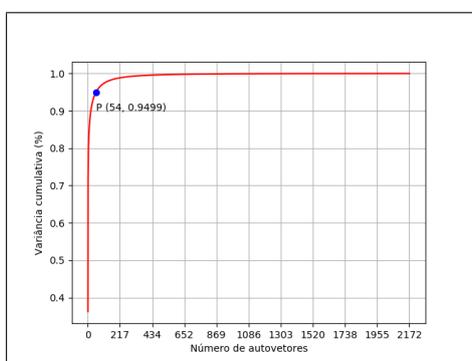
Apesar de não haver um consenso sobre a quantidade de componentes principais se deve manter, [Swets and Weng 1996] consideram que as melhores taxas de reconhecimento são obtidas ao se escolher uma quantidade de autovetores que correspondam à correspondem cerca de 95% da variância total. Portanto, para os experimentos utilizou-se uma quantidade de *eigenfaces* (componentes principais) que equivale 95% da variância total explicada, obtida através da soma cumulativa da variância explicada por cada componente principal. Para cada subconjunto de treinamento do método *Stratified k-fold* a variância cumulativa para todos os autovetores foi calculada para que fossem selecionados *eigenfaces* que representam os 95% de variância já explicados.

Na Figura 2, referente à primeira iteração do método *Stratified k-fold*, é possível observar que são necessárias apenas 177 autovetores dos 360 para se obter 95% da variância cumulativa. Dessa forma para essa iteração foram utilizadas 177 *eigenfaces* para se obter uma nova representação para as faces da base de dados AT&T presentes no conjunto de treinamento dessa iteração.



**Figura 2. Relação entre o número de componentes principais e a taxa de variação cumulativa explicada relativa ao subconjunto de treinamento da base de dados AT&T.**

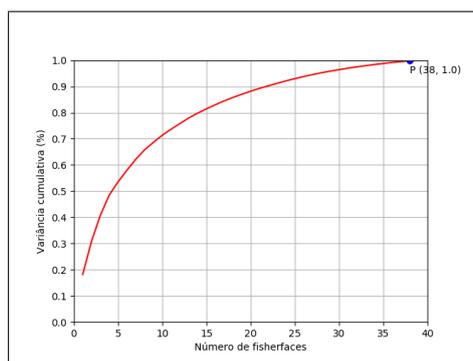
De forma análoga, para a base *Extended Yale B* escolhendo-se 95% da variância total explicada, temos o gráfico da variação cumulativa explicada na Figura 3. Note que para essa base de dados são necessárias 54 componentes principais para representar os 95% de variância total explicada.



**Figura 3. Relação entre o número de componentes principais e a taxa de variação cumulativa explicada relativa ao subconjunto de treinamento da base de dados Extended Yale B.**

## 2.2. Fisherfaces - LDA

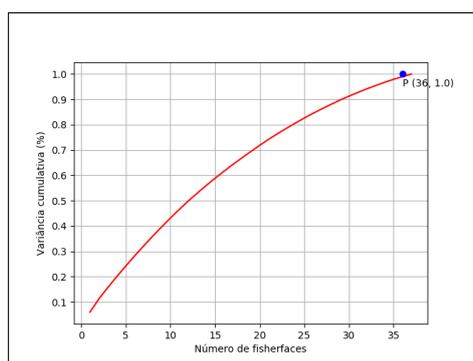
Similarmente ao que foi visto na Figura 2 para o método Eigenfaces, a variância cumulativa para todos os autovetores obtidos pelo subconjunto de treinamento foi calculada para o método Fisherfaces. Na Figura 4 é possível observar que são necessárias 38 autovetores (*fisherfaces*) para se obter 100% da variância cumulativa. Dessa forma, foram utilizadas 38 *fisherfaces* para se obter uma nova representação para as faces da base de dados AT&T na primeira iteração do método *stratified k-fold*.



**Figura 4. Relação entre o número de *fisherfaces* e a taxa de variação cumulativa explicada relativa ao subconjunto de treinamento da base de dados AT&T.**

Segundo [Li et al. 2010] e [Giron-Sierra 2016, p.806], o número máximo de *fisherfaces* válidas é igual a  $c - 1$ , onde  $c$  é o número de classes. Dessa forma, para o subconjunto de treinamento da base de dados AT&T teremos no máximo 39 *fisherfaces*, visto que existem 40 classes nesse subconjunto, representadas por 40 indivíduos distintos.

Já para a base de dados *Extended Yale B*, De forma análoga, escolhendo-se 100% da variância total explicada, temos o gráfico da variação cumulativa explicada na Figura 5.



**Figura 5. Relação entre o número de *fisherfaces* e a taxa de variação cumulativa explicada relativa ao subconjunto de treinamento da base de dados *Extended Yale B*.**

## 3. Experimentos

Para os experimentos 1 e 2 descritos à seguir foram utilizados os parâmetros mostrados na Tabela 1. Pelo fato de não existirem regras absolutas sobre que parâmetros utilizar buscou-se realizar os experimentos com as alternativas mostradas na já mencionada Tabela 1. Para

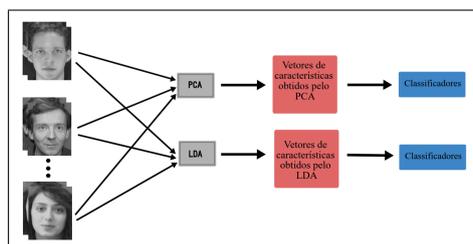
o classificador KNN variou-se o valor de  $k$  e a distância utilizada. Para o classificador SVM, variou-se a função kernel. Já para o classificador MLP variou-se a quantidade de neurônios e de camadas ocultas.

**Tabela 1. Parâmetros gerais utilizados nos experimentos.**

Classificador	Parâmetros	Valor
KNN	k	1; 3; 5; 7
	Distância	Euclidiana; Manhatan
SVM	Kernel	Linear; Polinomial
MLP	Taxa de aprendizagem	0.1
	Número máximo de épocas	1000
	Número de camadas escondidas	1; 2
	Momentum	0.3
	Solver	SGD
	Função de ativação	Logística (logistic)

Para o caso em que se testou a utilização de uma camada escondida, a quantidade de neurônios variou de 50 a 400 com um acréscimo de 50 neurônios entre cada teste. Já para o caso em que foram utilizadas duas camadas escondidas utilizou-se a seguinte notação: "x, y", na qual  $x$  representa o número de neurônios na primeira camada oculta e  $y$  a quantidade de neurônios na segunda camada oculta. Os resultados desses dois casos podem ser vistos nas Tabelas referentes à utilização do classificador MLP na seção Resultados.

Para o experimento 1, a base de dados AT&T foi utilizada, de forma que suas faces foram projetadas no espaço *eigenfaces* e no espaço *fisherfaces*. Os dados projetados no espaço *eigenfaces* foram utilizados para se treinar os três classificadores já mencionados. De forma análoga os dados projetados no espaço *fisherfaces* foram submetidos aos três classificadores mencionados. A Figura 6 faz uma representação do experimento 1.



**Figura 6. Modelo esquemático do experimento 1.**

Para o experimento 2, a base de dados *Extended Yale database B* foi utilizada, de forma que suas faces foram projetadas no espaço *eigenfaces* e no espaço *fisherfaces*. Os dados projetados no espaço *eigenfaces* foram utilizados para se treinar os três classificadores já mencionados. De forma análoga os dados projetados no espaço *fisherfaces* foram submetidos aos três classificadores mencionados.

Para a realização dos experimentos, foi utilizado um microcomputador com processador Intel Core i5-2450M, 2,5 Ghz e 6 GB de RAM, operando sob o sistema operaci-

onal Ubuntu 18.04.2 LTS. Para o desenvolvimento deste trabalho foi utilizada a biblioteca *Scikit-learn* [Pedregosa et al. 2011], desenvolvida em *Python* com propósito de auxiliar no estudo da aprendizagem de máquina. A versão da biblioteca *Scikit-learn* utilizada nos experimentos foi a versão 0.19.1, além disso foi utilizada a linguagem *Python* na versão 3.7.

#### 4. Resultados

Todos os resultados do experimento 1 são referentes à base de dados AT&T como foi mencionado na Seção 3. Na Tabela 2 são apresentados os resultados do classificador KNN treinado com os dados projetados utilizando a técnica Eigenfaces. De forma análoga, na Tabela 3 estão os resultados para a técnica Fisherfaces. Os resultados da acurácia média do classificador SVM, podem ser vistos nas Tabelas 4 e 5.

Para o treinamento do classificador MLP, foi utilizada apenas uma camada oculta, em um primeiro momento, e em um segundo momento foram utilizadas duas camadas ocultas. Para os testes do classificador MLP com uma camada escondida, a quantidade de neurônios variou de 50 a 400 com um acréscimo de 50 neurônios entre cada teste. Já para o caso de duas camadas, como no lado direito da Tabela 6, a quantidade de neurônios em cada uma das camadas é apresentada como números separados por vírgula. O resultado geral de ambos os casos podem ser vistos para cada uma das técnicas Eigenfaces e Fisherfaces nas Tabelas 6 e 7, respectivamente.

**Tabela 2. Resultados do método PCA + KNN no banco de imagens AT&T.**

Taxa de Acurácia (%) - PCA + KNN		
k	Distância Euclidiana	Distância Manhattan
1	<b>0.978 ± 0.021</b>	0.948 ± 0.042
3	0.957 ± 0.016	0.920 ± 0.043
5	0.930 ± 0.031	0.882 ± 0.039
7	0.897 ± 0.034	0.840 ± 0.036

**Tabela 3. Resultados do método LDA + KNN no banco de imagens AT&T.**

Taxa de Acurácia (%) - LDA + KNN		
k	Distância Euclidiana	Distância Manhattan
1	0.980 ± 0.024	<b>0.982 ± 0.025</b>
3	0.980 ± 0.022	0.972 ± 0.033
5	0.975 ± 0.022	0.980 ± 0.019
7	0.975 ± 0.025	<b>0.982 ± 0.020</b>

As Tabelas 8 – 13, referentes ao experimento 2, são análogas ao que foi visto no experimento 1, diferenciando-se apenas a base de dados utilizada, que no experimento 2, foi a *Extended Yale B*.

A Tabela 14 apresenta as maiores taxas de acurácia obtidas nos Experimentos 1 e 2. Para classificadores que obtiveram taxa de acurácia idênticas, foram escolhidos aqueles que possuíam menores valores de desvio padrão.

**Tabela 4. Resultados do método PCA + SVM no banco de imagens AT&T.**

<b>Taxa de Acurácia (%) - PCA + SVM</b>	
Kernel	Acurácia (%)
Linear	<b>0.982 ± 0.023</b>
Polinomial	0.970 ± 0.027

**Tabela 5. Resultados do método LDA + SVM no banco de imagens AT&T.**

<b>Taxa de Acurácia (%) - LDA + SVM</b>	
Kernel	Acurácia (%)
Linear	<b>0.985 ± 0.025</b>
Polinomial	0.960 ± 0.030

**Tabela 6. Resultados do método PCA + MLP no banco de imagens AT&T.**

<b>Taxa de Acurácia (%) - PCA + MLP</b>			
Neurônios	Acurácia	Neurônios	Acurácia
50	0.905 ± 0.057	50, 400	0.890 ± 0.032
100	0.963 ± 0.028	100, 350	0.938 ± 0.028
150	0.968 ± 0.022	150, 300	0.953 ± 0.021
200	0.975 ± 0.019	200, 250	0.975 ± 0.022
250	0.977 ± 0.018	250, 200	0.963 ± 0.032
300	<b>0.982 ± 0.016</b>	300, 150	<b>0.982 ± 0.020</b>
350	0.975 ± 0.019	350, 100	0.975 ± 0.019
400	0.975 ± 0.016	400, 50	0.967 ± 0.022

**Tabela 7. Resultados do método LDA + MLP no banco de imagens AT&T.**

<b>Taxa de Acurácia (%) - LDA + MLP</b>			
Neurônios	Acurácia	Neurônios	Acurácia
50	0.975 ± 0.030	50, 400	0.080 ± 0.024
100	0.978 ± 0.024	100, 350	0.065 ± 0.025
150	0.977 ± 0.021	150, 300	0.982 ± 0.023
200	0.985 ± 0.020	200, 250	0.975 ± 0.025
250	<b>0.988 ± 0.017</b>	250, 200	0.982 ± 0.023
300	0.982 ± 0.016	300, 150	0.980 ± 0.019
350	0.982 ± 0.016	350, 100	0.977 ± 0.021
400	0.985 ± 0.017	400, 50	0.970 ± 0.027

Para o banco de imagens AT&T, o classificador LDA + MLP apresentou o melhor desempenho em relação aos demais classificadores, obtendo uma acurácia de 98,8%, entretanto os demais classificadores também obtiveram resultados de acurácia acima de 97%, ou seja, também possuíram um excelente desempenho para os parâmetros selecio-

**Tabela 8. Resultados do método PCA + KNN no banco de imagens *Extended Yale B*.**

<b>Taxa de Acurácia (%) - PCA + KNN</b>		
k	Distância Euclidiana	Distância Manhattan
1	0.618 ± 0.261	<b>0.809 ± 0.234</b>
3	0.537 ± 0.258	0.790 ± 0.242
5	0.535 ± 0.272	0.780 ± 0.250
7	0.534 ± 0.278	0.777 ± 0.250

**Tabela 9. Resultados do método LDA + KNN no banco de imagens *Extended Yale B*.**

<b>Taxa de Acurácia (%) - LDA + KNN</b>		
k	Distância Euclidiana	Distância Manhattan
1	0.917 ± 0.123	0.919 ± 0.115
3	0.916 ± 0.123	0.918 ± 0.115
5	0.921 ± 0.119	0.921 ± 0.113
7	0.921 ± 0.118	<b>0.922 ± 0.115</b>

**Tabela 10. Resultados do método PCA + SVM no banco de imagens *Extended Yale B*.**

<b>Taxa de Acurácia (%) - PCA + SVM</b>	
Kernel	Acurácia (%)
Linear	<b>0.838 ± 0.181</b>
Polinomial	0.785 ± 0.210

**Tabela 11. Resultados do método LDA + SVM no banco de imagens *Extended Yale B*.**

<b>Taxa de Acurácia (%) - LDA + SVM</b>	
Kernel	Acurácia (%)
Linear	0.780 ± 0.102
Polinomial	<b>0.802 ± 0.143</b>

nados.

De forma geral, levando-se em consideração todos os resultados das tabelas apresentadas nesta seção podemos notar que, com relação ao banco de imagens AT&T os resultados para ambas as técnicas PCA e LDA obtiveram valores semelhantes para todos os classificadores apresentados, porém a técnica LDA obteve valores de acurácia ligeiramente melhores que a PCA.

Já para o banco de imagens *Extended Yale B*, o classificador LDA + MLP apresentou o melhor desempenho, com uma acurácia de 94,2% para uma MLP com uma camada

**Tabela 12. Resultados do método PCA + MLP no banco de imagens *Extended Yale B*.**

<b>Taxa de Acurácia (%) - PCA + MLP</b>			
Neurônios	Acurácia	Neurônios	Acurácia
50	0.204 ± 0.138	50, 400	0.285 ± 0.157
100	0.304 ± 0.153	100, 350	0.445 ± 0.217
150	0.425 ± 0.233	150, 300	0.476 ± 0.230
200	0.468 ± 0.252	200, 250	0.544 ± 0.261
250	0.518 ± 0.269	250, 200	0.566 ± 0.243
300	0.521 ± 0.284	300, 150	0.585 ± 0.255
350	0.559 ± 0.281	350, 100	0.608 ± 0.259
400	0.569 ± 0.266	400, 50	<b>0.611 ± 0.254</b>

**Tabela 13. Resultados do método LDA + MLP no banco de imagens *Extended Yale B*.**

<b>Taxa de Acurácia (%) - LDA + MLP</b>			
Neurônios	Acurácia	Neurônios	Acurácia
50	0.927 ± 0.117	50, 400	0.889 ± 0.135
100	0.935 ± 0.116	100, 350	0.916 ± 0.120
150	0.938 ± 0.101	150, 300	0.922 ± 0.122
200	0.938 ± 0.106	200, 250	0.924 ± 0.117
250	0.939 ± 0.108	250, 200	0.921 ± 0.133
300	0.941 ± 0.105	300, 150	0.920 ± 0.122
350	0.939 ± 0.106	350, 100	0.926 ± 0.112
400	<b>0.942 ± 0.105</b>	400, 50	0.909 ± 0.132

oculta constituída por 400 neurônios, como pode ser visto na Tabela 14. Em segundo lugar, o classificador LDA + KNN apresentou uma acurácia de 92,2% para  $k = 7$ . Os demais classificadores obtiveram um desempenho inferior, de modo que suas acurácias obtiveram valores inferiores à 84%.

Levando-se em consideração todos os resultados das Tabelas apresentadas nesta seção podemos notar que, para essa base de imagens, os classificadores que utilizaram a técnica Fisherfaces obtiveram melhores resultados para quase todos os classificadores estudados, exceto para o classificador SVM, que apresentou uma acurácia de 83,8% utilizando PCA e Kernel Linear em oposição aos 80,2% obtidos utilizando LDA e Kernel polinomial, como pode ser visto nas Tabelas 10 e 11, respectivamente. Em especial, o classificador MLP com a técnica Fisherfaces obteve resultados bastante superiores aos com a técnica Eigenfaces.

## 5. Conclusão e Trabalhos Futuros

De forma geral a técnica LDA obteve maiores valores de acurácia média do que a técnica PCA para a configuração de parâmetros mostrada na metodologia. Sob condições extremas de iluminação como as vistas na base de dados *Extended Yale Database B* o método

**Tabela 14. Melhores acurácias médias obtidas nos experimentos 1 e 2.**

Taxa de Acurácia (%) - LDA + MLP			
Base de dados	Classificador	Parâmetros	Acurácia média
AT&T	PCA + KNN	k = 1, distância Euclidiana	0.978
	LDA + KNN	k = 7, distância Manhattan	0.982
	PCA + SVM	kernel = linear	0.982
	LDA + SVM	kernel = linear	0.985
	PCA + MLP	nº neurônios = 300	0.982
	LDA + MLP	nº neurônios = 250	0.988
<i>Extended Yale B</i>	PCA + KNN	k = 1, distância Manhattan	0.809
	LDA + KNN	k = 7, distância Manhattan	0.922
	PCA + SVM	kernel = linear	0.838
	LDA + SVM	kernel = Polinomial(grau 2)	0.802
	PCA + MLP	nº neurônios = 400, 50	0.611
	LDA + MLP	nº neurônios = 400	0.942

LDA (Fisherfaces) mostrou-se bastante superior, em termos de acurácia média, ao método PCA (Eigenfaces).

Para a base de dados AT&T, que possui pequenas variações na posição do rosto, bem como pequenas variações de iluminação ambos os métodos obtiveram resultados equivalentes, pois suas diferenças de acurácias médias são mínimas.

Como trabalhos futuros sugere-se que sejam feitos testes para outras bases de dados de imagens com atributos diferentes dos vistos nas bases da AT&T e *Extended Yale B*. Além disso, sugere-se que sejam feitos testes com parâmetros diferentes dos que foram mostrados nesse trabalho para os classificadores utilizados. Além disso, sugere-se variar a porcentagem de componentes utilizadas para verificar a influência do número de componentes nos resultados finais.

Outros algoritmos de extração de características podem ser testados, como por exemplo, análise de componentes independentes (ICA), tanto para as bases de dados aqui apresentadas como para outras bases de dados, como JAFEE (*Japanese Female Facial Expression*) ou *Yale Face database*. Pode-se também combinar as técnicas de extração de características com outros classificadores não abrangidos nesse trabalho. Além disso, pode-se verificar o desempenho da combinação das técnicas PCA e LDA combinadas para a extração de características.

## Referências

AT&T Laboratories Cambridge (2002). The database of faces.

Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720.

Coppin, B. (2010). *Inteligencia Artificial*. LTC.

- Eleyan, A. and Demirel, H. (2007). Pca and lda based neural networks for human face recognition. In *Face Recognition*. InTech.
- Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660.
- Giron-Sierra, J. (2016). *Digital Signal Processing with Matlab Examples, Volume 2: Decomposition, Recovery, Data-Based Actions*. Signals and Communication Technology. Springer Singapore.
- Lee, K., Ho, J., and Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5):684–698.
- Li, X., Ma, S., and Irwin, G. W. (2010). *Life System Modeling and Intelligent Computing*. Springer.
- Martínez, A. M. and Kak, A. C. (2001). Pca versus lda. *IEEE transactions on pattern analysis and machine intelligence*, 23(2):228–233.
- Müller, A. and Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O’Reilly Media.
- Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Silva, A. L. and Cintra, M. E. (2015). Reconhecimento de padrões faciais: Um estudo. In *Encontro Nacional de Inteligência Artificial e Computacional, 2015, Proceedings ENIAC*, pages 224–231.
- Silva, G. N. d. (2009). Estudo da técnica pca (análise de componentes principais) e auto-faces aplicadas ao reconhecimento de faces humanas.
- Swets, D. L. and Weng, J. J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8):831–836.
- Tan, K.-H., Kriegman, D. J., and Ahuja, N. (2002). Appearance-based eye gaze estimation. In *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 191–195. IEEE.
- Toygar, Ö. and Adnan, A. (2003). Face recognition using pca, lda and ica approaches on colored images. *IU-Journal of Electrical & Electronics Engineering*, 3(1):735–743.
- Turk, M. and Pentland, A. (1991a). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86.
- Turk, M. A. and Pentland, A. P. (1991b). Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR’91., IEEE Computer Society Conference on*, pages 586–591. IEEE.
- Zhang, X. and Gao, Y. (2009). Face recognition across pose: A review. *Pattern Recognition*, 42(11):2876–2896.