



APLICAÇÃO DA MATRIZ DE FATORAÇÃO NÃO NEGATIVA PARA VISÃO COMPUTACIONAL

Trabalho de Conclusão de Curso

Engenharia da Computação

Bruno Henrique Massakazu Yamashita
Orientador: Prof. Bruno José Torres Fernandes



Universidade de Pernambuco

**Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação**

BRUNO HENRIQUE M. YAMASHITA

**APLICAÇÃO DA MATRIZ DE
FATORAÇÃO NÃO NEGATIVA PARA
VISÃO COMPUTACIONAL**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Novembro de 2013.

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)

No dia 19 de 12 de 2013, às 9:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente BRUNO HENRIQUE MASSAKAZU YAMASHITA, orientado pelo professor Bruno José Torres Fernandes, sob título APLICAÇÃO DA MATRIZ DE FATORAÇÃO NÃO NEGATIVA PARA VISÃO COMPUTACIONAL, a banca composta pelos professores:

Byron Leite Dantas Bezerra

Bruno José Torres Fernandes

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

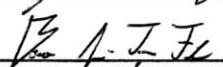
Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 9,0 (NOVE)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 04 dias para entrega da versão final da monografia a contar da data deste documento.


BYRON LEITE DANTAS BEZERRA


BRUNO JOSÉ TORRES FERNANDES

*Dedicado a minha família, noiva e amigos de universidade
por estes cinco anos de luta, estudo, esforço e dedicação.*

Agradecimentos

Mais um ciclo se encerra e um novo se inicia. É o que todos dizem. Minha vida na universidade foi muito gratificante em todos os sentidos, seja nas noites viradas, o cansaço, o estresse e principalmente a satisfação de aprender coisas novas e inovadoras. Acredito que nesses aspectos este curso de Engenharia de Computação da POLI, só fez saciar e aumentar minha ânsia por esta área que me impressiona e me leva a novos patamares.

Agradeço primeiramente a Deus, por ter me dado tantas felicidades e desafios para serem superados em meu cotidiano. A minha família por ter enfrentado todos meus problemas decorrentes da Universidade. Minha noiva Poliana pela paciência, dedicação e amor. E a todos meus amigos que fiz durante o andar do curso, todos são especiais.

Dentre esses amigos feitos na UPE, gostaria de agradecer especialmente ao Rodrigo Augusto, por todos os trabalhos em grupo que fizemos, que com certeza, mesmo com os prazos correndo e o tempo esgotando foram bastante divertidos de certa forma. Agradeço também a todos os alunos da turma que entraram no semestre de 2009.1, que tornaram minha vida muito mais alegre e descontraída durante esses cinco anos de curso, não vou citar nomes porque temo não lembrar todos.

Com relação ao projeto, agradeço ao meu orientador Bruno Fernandes, pela paciência e pelos conselhos dados, assim como a assistência durante todo o projeto (pode ter certeza que levarei o que aprendi a diante). Agradecimento especial para Carlos Castor, sua ajuda foi primordial para meu entendimento das técnicas abordadas aqui neste projeto, principalmente no momento inicial.

Fica aqui o meu obrigado!

Resumo

O Reconhecimento de faces é um problema desafiador em Visão Computacional. Diversos métodos com sofisticadas abordagens têm sido desenvolvidas a fim de obter melhores resultados utilizando dados de faces. Uma das técnicas utilizadas para realizar esta tarefa de reconhecimento é a Fatoração de Matrizes Não Negativas (NMF – *Non Negative Matrix Factorization*) sendo capaz de aprender partes de faces e semânticas de texto. Contrastando com outra técnica, conhecida como a Análise de Componentes Principais (PCA – *Principal Components Analysis*) e tendo como base os resultados obtidos por Lee, foi feita uma análise da NMF quanto ao seu desempenho, taxas de acerto, realizando paralelos com a técnica PCA para o banco de faces ORL.

Abstract

Face recognition is a challenging problem in computer vision. Several methods with sophisticated approaches have been developed in order to obtain better results using a face data. One of the techniques used to accomplish this task of recognition is the Non Negative Matrix Factorization (NMF) being able to learn parts of faces and semantic text. In contrast to other technique known as Principal Components Analysis (PCA) and based on the results obtained by Lee, it was taken an analysis of the NMF about your exploitation (hit rates) and performing parallel with the PCA for the ORL face database.

Sumário

Índice de Figuras	3
Índice de Tabelas	5
Tabela de Símbolos e Siglas	6
Capítulo 1 Introdução	7
1.1 Aplicações e contexto tecnológico.....	8
1.2 Objetivo	9
1.3 Organização do trabalho	10
Capítulo 2 Métodos para extração de características	11
2.1 Pré-processamento das técnicas	12
2.2 Restrições das técnicas impostas à fórmula geral.....	14
2.3 A Análise de Componentes Principais (PCA).....	15
2.4 Matriz de Fatoração Não Negativa (NMF).....	20
2.5 Resultado da reconstrução da face	23
Capítulo 3 Modelo de classificação e análise dos resultados	25
3.1 Classificador de distância mínima	26
3.2 Cálculo das imagens transformadas	27
3.3 Experimentos e resultados obtidos.....	27
Capítulo 4 Considerações finais	33
4.1 Conclusão	33
4.2 Trabalhos futuros.....	34
Referências Bibliográficas	35
Apêndice A Codificação do projeto em MATLAB	39

Índice de Figuras

1. Exemplos de faces retirados de Sinha [2]. Da esquerda para direita estão as pessoas: Príncipe Charles, Woody Allen, Bill Clinton, Saddam Hussein, Richard Nixon e Princesa Diana. 7
2. Base de dados gerada para teste do PCA no trabalho de Smith [10]. Do lado esquerdo é a base pura, do lado direito são os valores calculados subtraídos da média calculada na Seção 2.1 aplicado em cima de seus valores. 16
3. Matriz de covariância gerada no trabalho de Smith [10] 17
4. Autovalores e Autovetores gerados do trabalho de Smith [10] 17
5. Linha vermelha mostra a distribuição principal dos dados e a linha azul mostra a componente secundária. Imagem extraída de Smith [10] 18
6. Resultados para reconstruir a imagem de uma face utilizando a técnica PCA. Imagem retirada do trabalho de Lee [7]. R (Número de dimensões) = 49. Ao lado esquerdo a matriz de imagens base (W) e do lado direito a matriz de transformação (H) 19
7. Modelo em formato de rede da NMF proposto por Lee [7]. Ao lado esquerdo a estrutura em rede proposta. Ao lado direito de cima para baixo: mapeamento da intensidade da característica em H , mapeamento da característica em W e, por fim, a visualização da característica na imagem original. 22
8. Reconstrução de uma imagem utilizando a NMF. Imagem retirada do trabalho de Lee [7]. Com R (número de dimensões) = 49 23
9. Imagens reconstruídas. A imagem da esquerda é a imagem original, a do meio é a imagem reconstruída pela NMF e a da direita é a imagem reconstruída pela PCA 23
10. Processo utilizado no projeto para o reconhecimento das faces 28

Índice de Tabelas

3.1 Resultados dos algoritmos obtidos no reconhecimento de faces para o banco de faces ORL. Percentual de acerto com o valor do desvio padrão para 30 rodadas do experimento e redução de dimensão para o valor $R = 49$	28
---	----

Tabela de Símbolos e Siglas

(Dispostos em ordem alfabética)

1NN – *1-nearest Neighbor*

KNN – *K-nearest Neighbor*

LNMF – *Local Non Negative Matrix Factorization*

NMF – *Non Negative Matrix Factorization*

ORL – Banco de Dados de faces da Universidade de Cambridge

PCA – *Principal Components Analysis*

Capítulo 1

Introdução

O reconhecimento de faces é um dos mais desafiadores problemas na Visão Computacional, pois as imagens de faces apresentadas podem conter uma enorme variedade de características como: a iluminação, expressão facial, pose, distância, dentre outros [1]. Existe um número considerável de aplicações em potencial que fazem uso desta área promovendo a interação com o ser humano, na qual a face é utilizada como cadastro em diversos sites da *internet*.

Os seres humanos são exímios reconhecedores de faces. No trabalho de Sinha [2], ele trata exatamente sobre este tema de reconhecimento, fazendo paralelo entre os modelos matemáticos e a visão dos seres humanos. Ele demonstra que os seres humanos são capazes de detectar imagens bastante ruidosas e embaçadas, basta somente que aquelas pessoas analisadas sejam pessoas conhecidas para os observadores, como mostra na Figura 1.



Figura 1. Exemplos de faces retirados de Sinha [2]. Da esquerda para direita estão as pessoas: Príncipe Charles, Woody Allen, Bill Clinton, Saddam Hussein, Richard Nixon e Princesa Diana.

Note que, mesmo com um nível de degradação das faces, é possível reconhecer as pessoas presentes na Figura 1. O que torna a tarefa dos modelos matemáticos bastante complexa ao compararmos com o modelo visual humano [2], pois é necessário simular tal visão com equações e teoremas e isso não é trivial de ser feito.

Para se reconhecer uma face é importante saber diferenciar as duas etapas realizadas para este objetivo: o que é reconhecer [3] e o que é detectar [4] uma face. A primeira etapa para sistemas de reconhecimento necessita que haja uma delimitação de uma região da face [4], essa delimitação é chamada de detecção. É nesta etapa que vamos observar uma foto e saber se existe um rosto presente nela ou não. Depois de feita a detecção, tendo como premissa de que uma face foi detectada, somente então é feito o reconhecimento. Em outras palavras, após perceber que existe uma face naquela imagem, o objetivo é saber de quem, ou programaticamente falando, a qual classe ela pertence [3].

Apesar de ambas as etapas serem utilizadas sempre em conjunto, para este projeto de conclusão de curso, o foco do estudo ficou no reconhecimento da face. O próprio banco utilizado nos testes, conhecido como ORL [5], já possui imagens com a área de face delimitada, o que é possível abstrair a questão da detecção da face.

Para realizar o reconhecimento, duas técnicas foram abordadas neste trabalho, que trabalha diretamente com a extração de características das faces, tornando possível saber a qual pessoa (classe) aquela face pertence: A Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis* [6]) e a Matriz de Fatoração Não Negativa (NMF, do inglês *Non Negative Matrix Factorization* [7]).

1.1 Aplicações e contexto tecnológico

No âmbito de Visão Computacional, um fator bastante importante deve ser reconhecido: os equipamentos computacionais estão mais rápidos, o que segundo Lopes [8] permite avanços bastante visíveis nesta área. Não é difícil encontrar

aparelhos móveis, máquinas fotográficas ou até mesmo redes sociais (como o *Facebook* [9]) que já têm algum tipo de reconhecimento facial, seja para identificar uma pessoa, ou a expressão que ela está realizando [3].

São aplicações bem presentes no cotidiano das pessoas e de comum uso, seja para facilitar a colocação de uma legenda numa foto, ou que sirva como uma segurança extra a um dispositivo, ou ainda que seja de um sistema que monitore entradas e saídas de colaboradores de uma empresa [8].

Lopes [8] também afirma que esse crescimento é de extrema importância para esta área de pesquisa, pois o processamento dos aparelhos permite uma análise rápida permitindo a identificação de pessoas a partir de imagens de vídeos ou até de imagens estáticas. Essas aplicações vêm sendo utilizadas como sistemas de acesso a prédios e bancos, inclusive sendo acoplados em robôs que além da capacidade de reconhecimento verificam também o estado emocional de seus operadores [8]. O que demonstra cada vez mais a importância deste tipo de aplicação para o cotidiano das pessoas, o quanto eles as impacta e como são de grande valia para o mercado.

1.2 Objetivo

O intuito deste projeto é programar duas abordagens de reconhecimentos de padrões para faces. A primeira, proposta por Lee [7], na qual o autor propõe a utilização da NMF e é o foco deste trabalho de conclusão de curso, onde ele impõe restrições dos números de entradas serem não negativos. Em contrapartida, uma segunda técnica, a PCA, também citada por Lee [7] e contextualizada com mais detalhes por Smith [10] a fim de confrontar resultados e verificar para o banco de faces ORL [5] qual das técnicas possuem melhor taxa de acerto no reconhecimento destas imagens. Ao final, esta comparação mostra qual das técnicas obteve o melhor aproveitamento para acertar o reconhecimento.

1.3 Organização do trabalho

O texto deste trabalho se organiza de forma que o processo para cálculo das técnicas seja explicado de forma incremental e temporal. Por este motivo, no Capítulo 2, os seguintes tópicos foram abordados para realizar a extração das características das faces:

- O pré-processamento das técnicas NMF [7] e PCA [6]. Este pré-processamento é importante para evitar viés no resultado e será detalhado separadamente para cada técnica.
- O cálculo da matriz de transformação [6] [11]. Em ambas a abordagem é nesta matriz está presente a intensidade de cada característica observada pela estrutura.

Lee [7] explica que para ambas as abordagens NMF e PCA, a forma de cálculo é bastante similar, a maior diferença entre elas são somente as restrições impostas por cada uma, por este motivo, o Capítulo 2 vai tratar diretamente desta semelhança e das diferenças dentre as duas abordagens. Desde o início com o pré-processamento até a geração das características.

Por fim, o Capítulo 3 aborda detalhes para classificação do classificador 1NN [12], o cálculo das imagens transformadas e explicando como foi feito o processo para a classificação com detalhes de alguns dados estatísticos em relação ao número de experimentos feitos e as taxas de acerto de cada técnica.

Por fim, o Capítulo 4 mostra conclusões do trabalho com algumas observações e aborda possíveis temas de trabalhos futuros.

Capítulo 2

Métodos para extração de características

As técnicas NMF e PCA têm praticamente o mesmo intuito quando se trata de reconhecimento de padrões [7]. Neste capítulo, em especial, o objetivo é exatamente para mostrar estas similaridades e diferenças para tornar possível tal reconhecimento em cada uma das duas técnicas. Em seu trabalho, Lee [7] expõe a seguinte equação

$$V_{iu} \approx (WH)_{iu} = \sum_{a=1}^R W_{ia} H_{au} \quad , \quad (2.1)$$

a qual é uma descrição de ambas as técnicas, PCA [10] e NMF [7]. V é uma matriz que possui um conjunto de imagens na qual se deseja reconhecer, com dimensões $n \times m$, no qual n contém o valor do pixel da imagem e m é o número de imagens utilizadas, onde V é aproximadamente uma multiplicação das matrizes W e H . É importante notar que as imagens originais, contidas na matriz V , são imagens compostas em vetores, na qual cada coluna da matriz recebe uma imagem. Esta abordagem é válida para ambas as técnicas. Portanto, a primeira etapa é organizar estas imagens em forma de vetor, para desta forma, sejam armazenadas em cada coluna de V para realizar corretamente todo o restante do processo de classificação. Portanto, iterativamente, o valor de i é exatamente um pixel de característica descrito dentre as n analisadas e o valor de u vai até o número m de exemplos utilizados. W são as chamadas imagens-base no qual as características que estão sendo aprendidas ficam armazenadas e H é a matriz de transformação, esta responsável

por fazer o reconhecimento das faces [7] [13] por possuir a intensidade na qual cada característica mapeada em W é presente na imagem original. Outra variável que deve ser observada é a variável α . Nela, definimos o valor do número de dimensões R , ou de características, que serão analisadas para a extração das características. Quanto maior este número, maior o número de características que serão analisadas. Um fator importante das duas técnicas é que, ambas têm a possibilidade de realizar a compressão e reconhecimento, e esta redução de dimensões analisadas estão definidas pela variável R . Em Delac [14] essa abordagem fica bastante clara, ao tratar diretamente sobre o tema de compressão de imagens, mas o autor também faz um paralelo com o reconhecimento de faces e assim como Lee [7], ressalta que o número de dimensões a serem reduzidas é um parâmetro que teve atenção especial para que o valor dado a ele não prejudicasse a classificação. Em outras palavras, caso essa redução exclua alguma característica que seria primordial para o reconhecimento das faces, este fator diminui a taxa de acerto do algoritmo.

Quanto à similaridade e diferença entre o cálculo das abordagens utilizando a fórmula geral, o que as diferencia, são basicamente as restrições impostas por cada uma delas sobre fórmula geral (Equação 2.1). Estas restrições serão melhor detalhadas na seção 2.2.

2.1 Pré-processamento das técnicas

Antes de realizar as operações, é necessário fazer um pré-processamento das imagens. Nesta seção, estão descritos os aspectos de pré-processamento de ambas as técnicas, com suas diferenças de acordo com os trabalhos de Pearson [6] e Smith [10] do lado da PCA e Lee [7] [11] do lado da NMF.

Para o PCA, os passos para calcular o pré-processamento são:

- **Obter um banco de amostras:** Para poder realizar a classificação de qualquer tipo de padrão, primeiro é necessário ter uma base de dados ao qual sirva para a extração de características. Segundo Nogueira [15], em seu trabalho sobre reconhecimento de padrões, na natureza, estes padrões se

manifestam de várias formas e características, é válido ressaltar que isso vale tanto para o reconhecimento de face, como para qualquer outro tipo de reconhecimento de padrões. E para realizar a tal classificação, um banco de dados com exemplos é necessário. Para este projeto, como já foi dito anteriormente, foi utilizado o banco de faces ORL [5].

- **Selecionar conjunto de treinamento:** Para validar que a abordagem funciona, separa-se um conjunto de imagens do banco para o treinamento e outro conjunto para testes. A partir do conjunto de treinamento é que a extração das características é feita. O conjunto de testes será incluído na etapa final para podermos validar o funcionamento da técnica.
- **Calcular a média:** Depois de realizada a coleta de dados e separadas as imagens utilizadas para o treinamento, calcula-se a média das características. Este valor é calculado a partir dos valores das características de todas as imagens a fim de obter a variância entre elas. Esse cálculo é feito *pixel a pixel* em tons de cinza de acordo com a equação

$$\bar{X}_i = \frac{\sum_{j=1}^m V_{ij}}{m}, \quad (2.2)$$

onde i é a média para cada característica presente no banco e j é o índice da m imagens analisadas.

- **Subtrair pela média:** Calculada esta média, é feita a subtração deste valor com todas as respectivas características das imagens originais de acordo com a equação

$$ImgsPréProcessadas = V_i - \bar{X}_i, \quad (2.3)$$

sendo i o índice da linha de características da matriz V .

Como não existem restrições de entradas de valores para a PCA, segundo Smith [10] e Pearson [6] já é possível realizar os cálculos de covariância, autovetores e autovalores que serão melhores detalhados na Seção 2.3.

Por outro lado, a NMF de Lee [7] [11], aceita somente números estritamente positivos como valores de entrada, então este pré-processamento utilizando a média não é de serventia para esta técnica. O que Lee propõe em seus trabalhos [7] e [11],

também utilizado nos experimentos de Li [16] com a NMF, é uma normalização dos dados. Com os seguintes passos:

- **Obter um banco de amostras:** Mesmo processo descrito no PCA.
- **Selecionar conjunto de treinamento:** Mesmo processo descrito no PCA.
- **Normaliza os dados:** A normalização é feita dividindo o valor das características pelo valor máximo dentre os valores do conjunto de treino daquela característica, descrita pela seguinte equação

$$ImgsPreProcessadas = V_i / \max(V_i) \quad , \quad (2.4)$$

tal que i é o índice do vetor de característica analisado sobre o valor máximo deste mesmo vetor. Além disso, um tratamento que deve ser feito nesta etapa para a NMF para evitar valores iguais à zero, é somar um pequeno valor a todos os valores dos pixels do banco [7] antes da normalização para evitar que esses possíveis zeros que apareçam após o cálculo evitando o *underflow*.

Com isso, o banco estará pronto para receber a lógica de cálculo da NMF.

2.2 Restrições das técnicas impostas à fórmula geral

Como dito anteriormente, utilizando a Equação 2.1 em questão, as diferenças entre PCA e NMF estão nas restrições impostas àquela fórmula [7]. Depois de realizado todo o pré-processamento das imagens, podemos seguir para os conceitos gerais de ambas as técnicas.

Tendo em mente sempre a Equação 2.1, a restrição do PCA é de que as colunas de W sejam ortonormais e as linhas de H sejam ortogonais entre si [7]. Esta abordagem permite uma representação para cada face sendo uma aproximação por uma combinação linear de todas as imagens de base, ou *eigenfaces* [17]. O grande problema é que por conter valores negativos como entrada, a sua saída por vezes

não possuem uma interpretação visual muito intuitiva [7]. Este entendimento visual não intuitivo foi gerado e mais detalhado na Seção 2.3.

A NMF, por sua vez, não tem como premissa receber valores negativos em suas matrizes W e H . O que a torna muito mais intuitiva, pois a adição deste conceito acrescentado às combinações lineares de múltiplas imagens de base, para representar uma face tornam os resultados muito mais intuitivos visualmente [7] (será mais bem detalhada na Seção 2.4).

Uma vez que os pré-processamentos sejam realizados, é possível agora aplicar os conceitos de ambas as técnicas PCA e NMF para a fase de treinamento. É nesta fase que os algoritmos definem suas respectivas matrizes de transformação e que são responsáveis pela geração das imagens transformadas, estas por sua vez, são possuem as características para reconhecimento das faces, pois possuem os valores de características [13].

2.3 A Análise de Componentes Principais (PCA)

A Análise de Componentes Principais, proposta em 1991 por Pearson [6], também chamada de Transformada Discreta de *Karhunen-Loève* ou Transformada de *Hotelling* [18] tem como principal objetivo, similarmente a NMF [7]: analisar dados visando sua redução e eliminação de características não necessárias com escolhas de formas representativas dos dados a partir de combinações lineares das variáveis originais [7].

É um método que serve para identificar padrões entre as características dos dados, tendo como possibilidade seu uso quando os vetores têm muitas dimensões e a representação gráfica da base de dados não é trivial [10]. Todavia, Smith [10] e Vasconcelos [18] ressaltam que esta técnica também pode ser utilizada para exemplos mais simples, com dimensões menores, como o mostrado da Figura 5. Este exemplo foi retirado do trabalho de Smith [10], onde o autor deseja analisar uma base de dados com duas dimensões apresentada na Figura 2.

	x	y		x	y
	2.5	2.4		.69	.49
	0.5	0.7		-1.31	-1.21
	2.2	2.9		.39	.99
	1.9	2.2		.09	.29
Data =	3.1	3.0	DataAdjust =	1.29	1.09
	2.3	2.7		.49	.79
	2	1.6		.19	-.31
	1	1.1		-.81	-.81
	1.5	1.6		-.31	-.31
	1.1	0.9		-.71	-1.01

Figura 2. Base de dados gerada para teste do PCA no trabalho de Smith [10]. Do lado esquerdo é a base pura, do lado direito são os valores calculados subtraídos da média calculada na Seção 2.1 aplicado em cima de seus valores.

Feito o pré-processamento das imagens, já explicado na Seção 2.1, os resultados tratados são a tabela do lado direito da Figura 2. Com eles, podemos agora calcular a matriz de covariância. Smith [10] explica que a covariância é a medida melhor aplicada para o contexto do PCA. Baseada na variância, a covariância verifica a variação dos dados, porém ao contrário da primeira, ela é utilizada para análise de dados com mais de uma dimensão, na equação a seguir está descrita a covariância para duas dimensões

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X}) - (Y_i - \bar{Y})}{(n - 1)}, \quad (2.5)$$

onde X e Y são os valores processados obtidos do pré-processamento descrito do lado direito da Figura 3 e como os valores foram subtraídos pela média, os valores para \bar{X} e \bar{Y} é igual a 0. Na Figura 3 está descrito os resultados obtidos por Smith [10] na matriz de covariância para o exemplo no qual o autor criou.

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

Figura 3. Matriz de covariância gerada no trabalho de Smith [10].

Como a base de dados é bidimensional, esta matriz terá dimensões 2×2 , ela é quadrada, sendo possível calcular os autovalores e autovetores. A importância deles é de vital significância, pois nos dão uma informação crucial para ser utilizada no PCA que é justamente o grau de importância de um determinado dado calculado. Para calcular o autovalor da matriz de covariância utiliza-se a equação

$$\det(\lambda.I - A) = 0 \quad (2.6)$$

Onde λ é o autovalor, I é a matriz identidade e A para este caso a matriz de covariância. Com a Equação 2.6 é possível calcular os dois valores para os autovalores apresentados na Figura 4. Para calcular os autovetores utiliza-se a equação

$$A.\vec{v} = \lambda.\vec{v} \quad (2.7)$$

Onde A é a matriz de covariância, \vec{v} é o autovetor e λ são os autovalores calculados na Equação 2.6. Ou seja, para cada autovalor, teremos um autovetor associado. Na Figura 4 mostra-nos os resultados obtidos do cálculo dos autovalores e autovetores utilizando como parâmetro da matriz de covariância apresentada na Figura 3.

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

Figura 4. Autovalores e Autovetores gerados do trabalho de Smith [10].

Os resultados, como podemos ver estão apresentados na Figura 5, onde os dados pré-processados estão plotados juntos com as duas componentes geradas pelos autovetores.

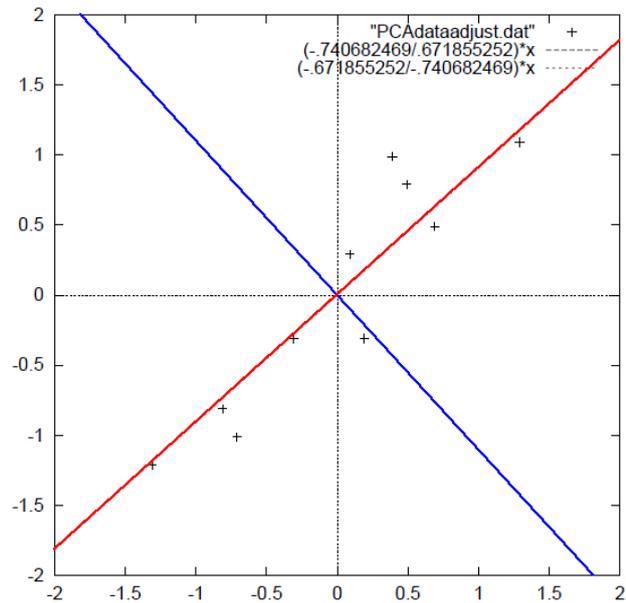


Figura 5. Linha vermelha mostra a distribuição principal dos dados e a linha azul mostra a componente secundária. Imagem extraída de Smith [10].

Note que, na Figura 5, dentre as retas componentes, que ao realizarmos uma projeção dos dados em cima da componente vermelha, ela concentraria melhores informações sobre os dados transformados.

Smith [10] explica que matematicamente, essa melhor aproximação se deve ao valor apresentado pelo autovalor. Quanto maior o autovalor, mais significativos são os dados daquele autovetor. O autor ainda explica que num contexto de compressão, quanto menor o valor deste autovalor menos significativo será os valores daquele autovetor para a representação daquela imagem. Então, caso optássemos por ignorar a componente azul que tem menor autovalor, a perda de dados significativos seria bem menor do que escolhermos somente a componente vermelha.

Porém, para um contexto de reconhecimento de face real, existem muito mais que duas dimensões, a matriz de transformação que será utilizada para realizar a classificação pode necessitar de mais dimensões que são definidas pelos R autovetores que estão associados com os R maiores autovalores, onde R é o número de dimensões que desejamos utilizar descrito na Equação 2.1. Na Figura 6, Lee [7] demonstra um resultado para o cálculo da PCA. Do lado esquerdo as imagens de base (matriz W na notação da Equação 2.1) e a matriz de transformação (matriz H na notação da Equação 2.1).

Note que os pixels em vermelho gerado na Figura 6, são valores os valores negativos apresentados como resultado da técnica. Podemos perceber que visualmente alguns padrões das imagens de base não são muito intuitivos. Não podemos saber exatamente quais partes de faces foram aprendidas por cada componente da PCA. Esse fator, segundo Lee [7], dá um grande impulso ao uso e estudo da NMF.

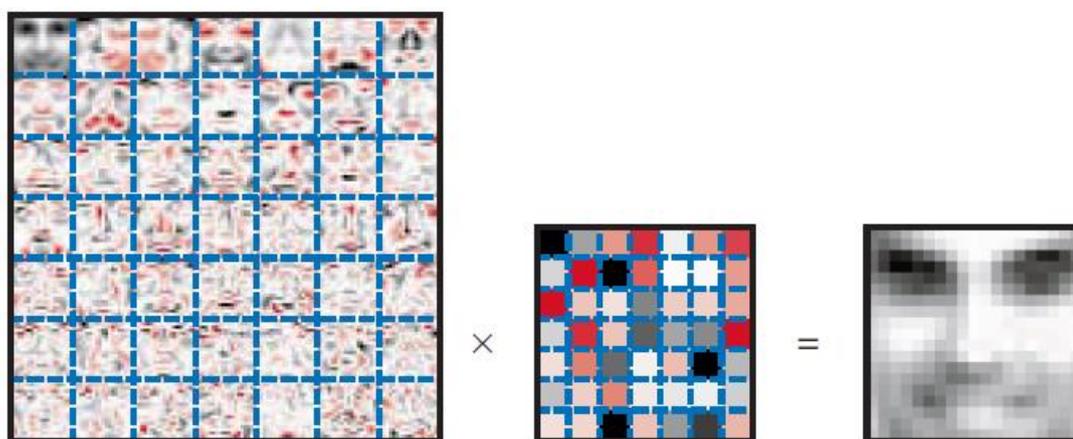


Figura 6. Resultados para reconstruir a imagem de uma face utilizando a técnica PCA. Imagem retirada do trabalho de Lee [7]. R (Número de dimensões) = 49. Ao lado esquerdo a matriz de imagens base (W) e do lado direito a matriz de transformação (H).

2.4 Matriz de Fatoração Não Negativa (NMF)

A Matriz de Fatoração Não Negativa (NMF), proposta por Lee [7], são um grupo de algoritmos, no qual uma matriz tem somente como entrada elementos não negativos. Por conta desta característica, ele demonstra em seu trabalho que são matrizes mais fáceis de analisar, por possuir somente valores positivos. É possível analisar qual parte de faces foram aprendidas na fase de treino da NMF, como sobrancelhas mais acentuadas, nariz mais largo, ouvidos, olhos, etc.

Em seu trabalho, Lee [7] utilizou a NMF para reconhecimento de face e de semântica de texto. Todavia, existem outros trabalhos que utilizam a sua ideia da NMF implementados, como: classificação de tumores [19], reconhecimento de ruídos em falas [20], separação de sinais musicais [21], estudo de eletroencefalogramas [22], dentre muitas outras. Em todos estes trabalhos, os autores valorizam o fato da restrição não negativa imposta pela premissa da NMF como uma excelente característica para reconhecimento de padrões pela sua maior intuitividade.

Depois de feito o pré-processamento das imagens realizado no Capítulo 2, é hora de calcular as matrizes W e H com as regras de atualização da NMF propostas por Lee [7] [11], dadas pelas equações

$$W_{ia} = W_{ia} \sum_{\mu} \frac{V_{i\mu}}{(WH)_{i\mu}} H_{a\mu} \quad , \quad (2.8)$$

$$H_{a\mu} = H_{a\mu} \sum_i W_{ia} \frac{V_{i\mu}}{(WH)_{i\mu}} \quad , \quad (2.9)$$

onde i é a característica que está sendo analisada, a é a dimensão dessa característica e μ é o índice do exemplo do conjunto de treinamento. Lee [7] explica que a convergência das atualizações iterativas pode ser provada pelo quociente $V_{i\mu} / (WH)_{i\mu}$, a fim de que WH seja uma aproximação, este valor fique bem próximo de 1. É importante notar que os valores da NMF [7] não são estocásticos como da PCA [6], a cada iteração do algoritmo, novos valores para W e H são atualizados com o decorrer de sua execução. Portanto, para realizar estas iterações,

primeiramente, iniciam-se com valores aleatórios de ambas as matrizes e esta inicialização é chamada de estado inicial da NMF [7]. É importante ressaltar que estes valores aleatórios influem com relação à inicialização da NMF, em outras palavras, os parâmetros salvos nas matrizes W e H nunca serão os mesmos a cada rodada de execução, diferentemente do PCA, que caso seja fixado uma base de treinamento para as rodadas de execução o resultado é o mesmo. Além disso, para conseguir realizar esta aproximação de $V \approx WH$ utiliza-se a equação

$$F = \sum_{i=1}^n \sum_{u=1}^m [v_{iu} \log(WH)_{ui} - (WH)_{ui}] \quad , \quad (2.10)$$

nomeada de função objetivo [7], a Equação 3.1 serve como ponto de parada do algoritmo. Ela é baseada na teoria de entropia relativa [11], na qual para a amplificação dos resultados quanto ao acerto do reconhecimento deve-se a um ponto de máximo local desta função, este ponto que depende de como ocorreu a inicialização da NMF. Lee [7] explica que o principal fator do sucesso da NMF, não é necessariamente esta equação. Além do uso da função objetivo, o ponto de parada pode ser calculado utilizando uma média de erro quadrado ao invés de utilizar esta função. Porém, este processo, por ser empírico, fazendo a multiplicação da imagem base com o respectivo valor da matriz de transformação (H) para verificar o acerto comparando com a imagem original, torna este processo mais demorado, dificultando bastante na fase de treino. Na Figura 7 expõe toda a estrutura de cálculo da NMF em forma de rede para um melhor entendimento de como é feita a reconstrução da imagem.

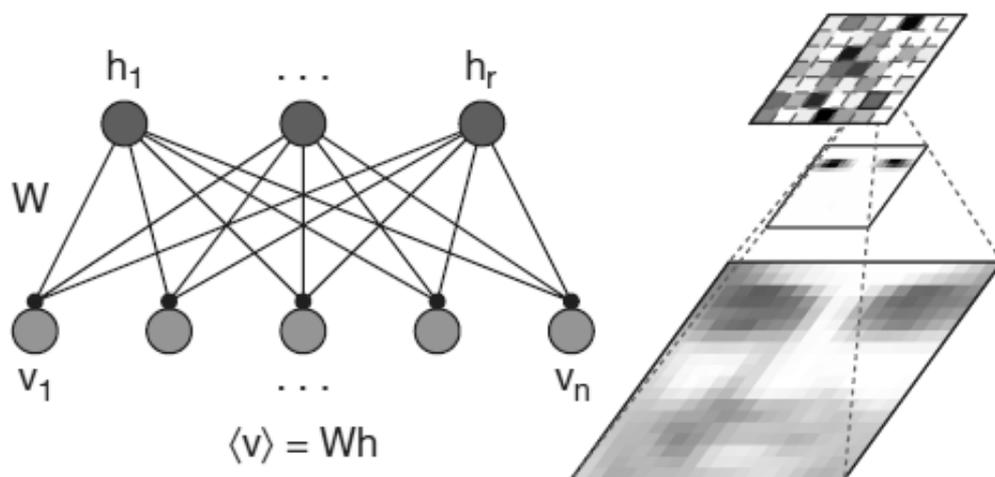


Figura 7. Modelo em formato de rede da NMF proposto por Lee [7]. Ao lado esquerdo a estrutura em rede proposta. Ao lado direito de cima para baixo: mapeamento da intensidade da característica em H , mapeamento da característica em W e, por fim, a visualização da característica na imagem original.

Assim, como na PCA, o autor também realiza uma reconstrução da imagem utilizando a NMF demonstrada na Figura 8. É possível notar que, em cada dimensão da matriz das imagens base W (matriz do lado esquerdo) as partes de faces que foram aprendidas em cada quadrante são muito mais intuitivas que na PCA. Podemos ver um pedaço de nariz, olhos, sobrancelhas, entre outras partes de face sendo identificadas nesta matriz. No meio da Figura 8, a matriz de transformação H , que similarmente a PCA, é a matriz com a intensidade das características e que será utilizada para o reconhecimento no próximo passo. Por fim, do lado direito é a face reconstruída pela multiplicação das duas estruturas.

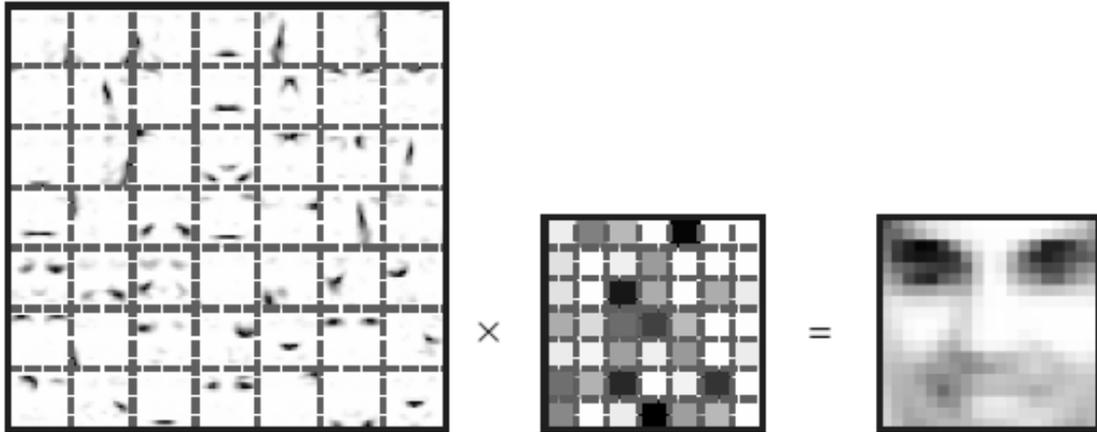


Figura 8. Reconstrução de uma imagem utilizando a NMF. Imagem retirada do trabalho de Lee [7]. Com R (número de dimensões) = 49.

2.5 Resultado da reconstrução da face

É possível notar que, mesmo com as reduções de dimensionalidade que as imagens sofrem as resultantes obtidas com ambas as técnicas são bastante satisfatórias ao observar a Figura 9. As diferenças são praticamente imperceptíveis, somente realizando uma subtração matemática das imagens para detectá-las.



Figura 9. Imagens reconstruídas. A imagem da esquerda é a imagem original, a do meio é a imagem reconstruída pela NMF e a da direita é a imagem reconstruída pela PCA.

Feito todo este processo para o cálculo da matriz de transformação, agora é possível realizar o reconhecimento das faces. Tendo em mente que este processo somente é calculado para uma parte do conjunto de faces (chamado grupo de

treinamento) que será detalhado no Capítulo 3. O papel do classificador vai ser de tendo a matriz de transformação calculada, calcular as imagens transformadas utilizando esta matriz para assim, identificar a qual grupo uma face (não presente no conjunto de treinamento) pertence.

Capítulo 3

Modelo de classificação e análise dos resultados

O reconhecimento de faces humanas é uma atividade rotineira no cotidiano nas pessoas [2]. Identificar aquela pessoa conhecida, por cruzar com ela numa rua, ou até saber o seu estado emocional na qual se encontra, este tipo de reconhecimento é fundamental [23]. Imaginando um mundo onde o ser humano não pudesse reconhecer uma pessoa, ou não notar em qual estado ela se encontra tornaria as relações sociais e comunicativas muito mais complicadas. É um tipo de mecanismo, na qual todas as pessoas desenvolvem desde muito cedo quando crianças e que com o tempo vai amadurecendo e culminando uma espantosa ferramenta de reconhecimento [23].

Com as matrizes de transformação calculadas, onde temos as intensidades de características de cada face de treinamento, podemos partir agora para a etapa final do projeto, que é justamente a classificação e reconhecer aquela pessoa. Khan [24] fez um trabalho parecido com o que foi desenvolvido neste projeto, porém, para reconhecimento de padrões quaisquer, utilizando somente a PCA. A mesma lógica de classificação pode também ser utilizada para a NMF. Ele demonstra o processo de classificação de padrões utilizando o mesmo algoritmo utilizado aqui neste projeto: o classificador de distância mínima (1NN, do inglês *1-Nearest Neighbor*). Neste capítulo, o objetivo é dar uma visão geral sobre o 1NN, demonstrar alguns exemplos do banco de faces ORL e explicar detalhes da metodologia de como se faz a fase de teste comentando os resultados obtidos.

3.1 Classificador de distância mínima

O classificador de distância mínima, também chamado de 1NN, é uma variante do KNN (do inglês, *K nearest neighbors* ou K vizinhos mais próximos). Esta técnica é simples, muito eficiente e efetiva para o campo de reconhecimento de padrões [12]. O objetivo dele é, uma vez que temos um dado de uma base conhecida, podemos utilizar esta técnica para dizer a qual classe ela pertence. Neste projeto, entendendo classe como uma pessoa e que cada classe tem dez exemplos (dez faces) podemos saber de qual classe aquela face faz parte. Em seu trabalho, Bhathia [12] descreve os mais diferenciados tipos de classificação utilizando a ideia central do KNN.

A técnica KNN já foi amplamente utilizada em vários campos: em reconhecimento de padrões nos trabalhos de Vaidehi [25] e Shizen [26], categorização de texto em Toker [27] e reconhecimento de objetos em Bajramovic [28]. Em outras palavras, quando se fala de reconhecimento de padrões, seja ele qual for sempre haverá a possibilidade de aplicação desta técnica para classificação.

Dos diferentes classificadores KNN existentes, foi decidido utilizar o 1NN. Ele é baseado no cálculo de distância euclidiana, que é bastante simples e rápido de se programar.

Basicamente, calcula-se a distância euclidiana (Equação 3.1) de um elemento de teste (que foi separado antes do treinamento) em cima das características obtidas nos elementos de treinamento. No exemplo deste projeto, teríamos a como uma sequência de pixels de uma imagem e b como outra sequência de pixels de outra imagem. Calculando esta distância entre elas, poderemos saber o quão parecido elas são.

$$D(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (3.1)$$

Como o 1NN é uma variação do KNN, significa que, dado que temos uma imagem transformada com as características dela, comparando com as características das outras imagens (presentes no banco de dados de treinamento) se aquela imagem X de teste é mais similar com uma imagem Y do treinamento e Y está associada a uma determinada classe Z, então a imagem de teste X será classificada como classe Z. Existe também a abordagem de um $K > 1$, que é o caso de pegarmos as K imagens de treino mais parecidas, e a classe que mais aparecer dentre estas imagens vizinhas é dada para aquela imagem. Esta abordagem para o valor de $K > 1$ não foi utilizada neste projeto.

3.2 Cálculo das imagens transformadas

Em seus trabalhos, Khan [24] e Nilangshu [29] realizam reconhecimento de padrões em faces utilizando a PCA. Neste projeto um procedimento parecido foi aplicado de acordo com estes dois trabalhos, refazendo os resultados da PCA e também os gerando para a NMF de Lee [7] [11]. Mas antes de realizarmos a classificação com o método NN, é necessário calcular as imagens transformadas, que são uma combinação linear da matriz de transformação H da Equação 2.1 com as imagens originais V do banco, como demonstrado na equação

$$It = H.V \quad , \quad (3.2)$$

onde H é a matriz de transformação e V o conjunto das imagens originais. É importante notar que estas imagens terão valores importantes de características de cada face, o que as torna propícias para compararmos suas características. Feito o cálculo das imagens transformadas It, agora podemos realizar a classificação das imagens com o método de classificação 1NN.

3.3 Experimentos e resultados obtidos

Para os experimentos foi utilizado o *software* aplicativo Matlab [30] por ser uma ferramenta de fácil utilização, não só para aplicações de imagens e Visão

Computacional como também por possuir um grande *framework* matemático para matrizes, o que ajudou bastante no processo de desenvolvimento das duas técnicas. Como já dito anteriormente, a PCA [6] e a NMF [7] são técnicas que têm como base a mesma fórmula geral (Equação 2.1). Portanto o processo para se reconhecer uma face utilizando estas técnicas é basicamente o mesmo, tendo como diferenças apenas, as restrições impostas por cada uma na fórmula geral. É importante notar que desde o início, este trabalho seguiu toda uma ordem cronológica para o reconhecimento de faces. Na Figura 10 podemos ver toda uma síntese do que foi descrito anteriormente cronologicamente e que foi o processo utilizado para ser possível classificar as faces dentre todas as classes presentes no banco.

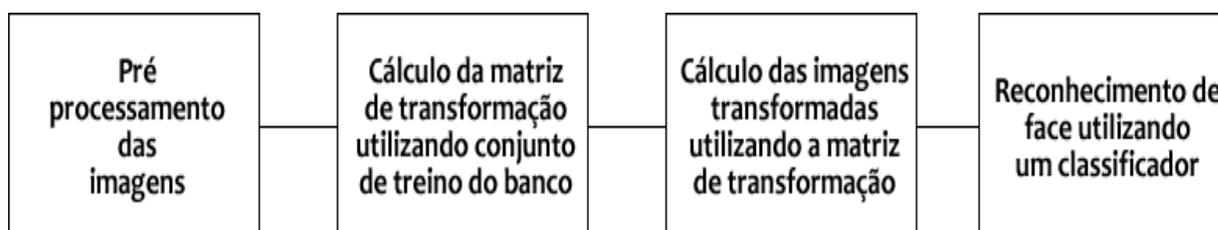


Figura 10. Processo utilizado no projeto para o reconhecimento das faces.

O banco de faces utilizado para realizar todos os experimentos foi o banco de faces da Universidade de Cambridge, o ORL [5]. Dentre suas características, ao total são 40 pessoas (classes), cada uma delas com 10 faces com diferentes expressões (olhos abertos / fechados, sorrindo / não sorrindo), iluminação, detalhes faciais (óculos e sem óculos) e posicionamento do rosto (para esquerda, para direita ou para frente). Na Figura 11 estão expostos alguns exemplos deste banco, que é utilizado como entrada de dados para os experimentos desta pesquisa.

Além disso, foram realizadas 30 rodadas de simulação para ambas as técnicas NMF e PCA com intuito de verificar a validade de ambas as técnicas e variando o número de imagens utilizadas para treino a fim de verificar a constância

de ambas nestes cenários. O número de dimensões utilizadas foi $R = 49$ que foi o parâmetro utilizado por Lee [7].



Figura 11. Exemplos de faces do Banco de faces ORL [5].

Abaixo, está descrito o número de exemplos que foi separado aleatoriamente em cada uma das 30 rodadas durante fase de treino para cada classe no treinamento das técnicas PCA e NMF:

- 70% das imagens utilizadas para treino e 30% das imagens utilizadas para teste.
- 50% das imagens utilizadas para treino e 50% das imagens utilizadas para teste.
- 30% das imagens utilizadas para treino e 70% das imagens utilizadas para teste.

Então, com as imagens transformadas (I_t) calculadas para o conjunto de treinamento e de teste, utilizando o conceito da Equação 3.2, é possível realizar a classificação 1NN para verificar a qual classe cada imagem de teste pertence.

Os resultados estão apresentados na Tabela,

Tabela 3.1 Resultados dos algoritmos obtidos no reconhecimento de faces para o banco de faces ORL. Percentual de acerto com o valor do desvio padrão para 30 rodadas do experimento e redução de dimensão para o valor $R = 49$.

Técnica Utilizada	Percentual de imagens para treino	30%	50%	70%
	Non negative Matrix Factorization (NMF)		71.64% (1.25%)	90,52% (1.21%)
Principal Component Analysis (PCA)		74.69% (1.23%)	89.50% (1.13%)	93.11% (1.26)

No qual podemos ver que, comparando as taxas de acerto das duas técnicas, os resultados foram bastante parecidos. O que mostra que a técnica NMF proposta por Lee [7] conseguiu resultados bem compatíveis com a PCA de Pearson [6], somente

perdendo no teste em que só foram utilizadas 30% das imagens de faces para treinamento, isso se deve ao posicionamento das componentes principais que deve ter separado melhor os agrupamentos quando utilizada com menos imagens de treino.

Capítulo 4

Considerações finais

4.1 Conclusão

O objetivo deste estudo era o aprendizado da técnica NMF para reconhecimento de padrões em Visão Computacional. Com estes resultados obtidos, fica claro que a utilização dessa técnica é aplicável para o reconhecimento de padrões.

Sabendo que num total poderiam ser incluídas todas as características na redução de dimensão, deixar que este valor utilize somente 49 dimensões, de um total de 92 x 112 (dimensões da imagem do banco ORL), com um acerto tão alto, demonstra o potencial que ambas as técnicas PCA e NMF possuem, tanto em reconhecimento de padrões como em compressão. É algo que vale a pena ser utilizado e estudado.

Em relação à NMF, o fato de uma técnica que se utilize apenas de valores positivos é de bom cunho para sua utilização no reconhecimento de padrões. Essas técnicas dão uma forma mais intuitiva de observar o aprendizado na fase de treinamento, pois permitem saber o que foi aprendido por cada parte daquela estrutura.

Certamente, em um ponto de vista de pesquisa, conhecer e aprender sobre estas técnicas para reconhecimento foi de grande proveito e conhecimento.

4.2 Trabalhos futuros

Durante os estudos para elaboração deste projeto final de curso, muitos foram os trabalhos encontrados sobre a NMF num geral. Existem muitas variações delas com diversas particularidades e diferenças. No trabalho de Li [16], por exemplo, além de citar a NMF proposta por Lee [7], ele também propõe uma nova abordagem chamada de Matriz de Fatoração Não Negativa Local (ou LNMF, do inglês *Local Non Negative Matrix Factorization*) onde ele realiza um aprendizado localizado, em uma parte de um subespaço para a representação de padrões visuais.

Existe também outra abordagem que utiliza a NMF como forma de processamento para entrada de Redes Neurais, que é o caso do trabalho de Barman [31] onde ele faz um aprendizado supervisionado de uma rede neural Perceptron e a sua primeira camada utiliza o conceito da NMF.

Outro aspecto, diante do trabalho do próprio Lee [7], é que ele também utiliza a NMF para reconhecimento de semânticas de texto, no exemplo ele utiliza a Carta de Independência dos Estados Unidos como dado de entrada.

Outra possibilidade é a utilização de uma abordagem híbrida utilizando a PCA e a NMF conjuntamente observando se, de fato, a quantidade de imagens para treino ou de outros parâmetros, como a redução de dimensão se comportam melhor em uma ou em outra técnica.

Todos estes estudos refletem a importância da técnica NMF em vários ramos de reconhecimento de padrões poderiam ser estudados e programados em trabalhos futuros.

Referências Bibliográficas

1. WANG, Y.; JIA, Y. **NON-NEGATIVE MATRIX FACTORIZATION FRAMEWORK FOR FACE RECOGNITION**. Beijing, China.
2. SINHA, P. et al. **Face Recognition by Humans**. Instituto de Tecnologia de Massachusetts. Cambridge. 2003.
3. HUANG, T.; ZIYOU, X.; ZHENQIU, Z. **Face Recognition Applications**. Universidade de Illinois. Springer New York, p. 371-390. 2005. (978-0-387-40595-7).
4. HJELMAS, E. **Face Detection: A Survey**. University of Oslo. Oslo, Norway, p. 236–274. 2001.
5. ORL. **Face Database**. Laboratório de computação da Universidade de Cambridge. Cambridge. 1994.
6. PEARSON, K. On Lines and Planes of Closest Fit to Systems of Points in Space. **Philosophical Magazine 2**, v. 2, p. 559-572, 1901.
7. LEE, D. D. Learning the parts of objects by non-negative matrix factorization. **Letters to Nature**, New Jersey, USA, p. 788-791, February 1999. ISSN 07974.
8. LOPES, E. C. **Detecção de Faces e Características Faciais**. Pontifícia Universidade Católica do Rio Grande do Sul. Porto Alegre, Rio Grande do Sul, Brasil. 2009.
9. FACEBOOK. Disponível em: <<http://www.facebook.com/>>.

10. SMITH, L. I. **A tutorial on Principal Component Analysis.** Universidade de Otago. Otago, Nova Zelândia. 2002.
11. LEE, D. D.; SEUNG, H. S. **Algorithms for Non-negative Matrix Factorization.** Instituto de Tecnologia de Massachusetts. Cambridge. 2001.
12. BHATIA, N.; V, V. **Survey of Nearest Neighbor Techniques.** International Journal of Computer Science and Information Security. Jalandhar, Índia, p. 302-305. 2010. (1947-5500).
13. GUILLAMET, D. **Classifying Faces with Non-negative Matrix Factorization.** Universidade de Barcelona. Barcelona, Spain, p. 1-8. 2010.
14. DELAC, K.; GRGIC, S.; GRGIC, M. **Image Compression in Face Recognition.** Universidade de Zagreb. Croatia, p. 1-15. 2008. (978-953-7619-34-3).
15. NOGUEIRA, A. A.; BAPTISTA, V.; SIQUEIRA, S. **Um Overview Sobre Reconhecimento de Padrões.** Associação Educacional Dom Bosco. Resende, Rio de Janeiro. 2008.
16. LI, S. Z. et al. **Local Non-Negative Matrix Factorization as a Visual Representation.** IEEE. Beijing, China. 2002. (0-7695-1459-6/02).
17. TURK, M.; PENTLAND, A. **Eigenfaces for recognition.** Congresso de Neurociência de Massachusetts. Instituto de Tecnologia de Massachusetts, p. 71-86. 1991.
18. VASCONCELOS, S. **Análise de Componentes Principais.** São

Paulo, Brasil. 2002.

19. YUEHUI, C.; XIFENG, X.; JINGRU, X. **Classifier Based on Non-negative Matrix Factorization for Tumor Data Classification.** IEEE. China, p. 936-940. 2010.
20. SCHULLER, B. et al. **NON-NEGATIVE MATRIX FACTORIZATION AS NOISE-ROBUST FEATURE EXTRACTOR FOR SPEECH RECOGNITION.** IEEE, Instituto para Comunicação Homem-Máquina. Munick, Alemanha, p. 4562-4569. 2010. (978-1-4244-4296-6).
21. TYGEL, A. F. **Métodos de Fatoração de Matrizes Não Negativas para separação de sinais musicais.** Rio de Janeiro, Brasil. 2009.
22. MINGYU, L.; WANG, J.; ZHENG, C. **Non-Negative Matrix Factorizations of Spontaneous Electroencephalographic Signals for Classification.** IEEE. China, p. 2790-2793. 2005.
23. ASCENSO, J.; VALENTIM, J.; PEREIRA, F. **RECONHECIMENTO AUTOMÁTICO DE FACES USANDO INFORMAÇÃO DE TEXTURA E DE GEOMETRIA 3D.** Instituto Superior Técnico das Telecomunicações. Lisboa. 2004. (1049-001).
24. KHAN, A.; FAROOQ, H. **Principal Component Analysis-Linear Discriminant Analysis Feature Extractor for Pattern Recognition.** IJCSI International Journal of Computer Science Issues. Punjab, Paquistão, p. 267-270. 2011. (1694-0814).
25. VAIDEHI, V.; VASUHI, S. **Person Authentication using Face Recognition.** Proceedings of the world congress on engg and

computer science. New York. 2008.

26. SHIZEN; WU, Y. An Algorithm for Remote Sensing Image Classification. **Springer Berlin**, Berlim, Alemanha, v. 40.
27. TOKER, G.; KIRMEMIS, O. **Text Categorization using k Nearest Neighbor**. Middle East Technical University. Survey Paper. 2008.
28. BAJRAMOVIE, F. **A Comparison of Nearest Neighbor Search**. ACIVS. California, San Diego, p. 1186-1197. 2006. (978-3-540-44632-3).
29. NILANGSHU, B. Pattern Recognition using Principal Component Analysis. **Binary Digits**, 31 August 2010. ISSN 82193801-9. Disponivel em: <<https://sites.google.com/site/binarydigits10/>>. Acesso em: 12 Outubro 2013.
30. MATHWORKS. **MathWorks**, 2009. Disponivel em: <<http://www.mathworks.com/>>. Acesso em: 02 dez. 2013.
31. BARMAN, P. C.; LEE, S.-Y. **Document Classification with Unsupervised Nonnegative Matrix Factorization and Supervised Perceptron Learning**. International Conference on Information Acquisition. Jeju, China, p. 182-186. 2007.

Apêndice A

Codificação do projeto em MATLAB

Abaixo, o código do projeto programado em Matlab:

```
% classificação do banco de faces ORL
function orl_classification

    nmfStr = 'NMF';
    pcaStr = 'PCA';

    % definir aqui se deseja utilizar a PCA [Pearson] ou
    % a PCA de [Lee]
    typeOfClassification = nmfStr;

    load('orl_experiment_7');

    train_data=orl_training_7; % banco de treino
    train_labels=orl_training_7_labels; % ID's do banco de treino
    test_data=orl_test_7; % banco de teste
    test_labels=orl_test_7_labels; % ID's do banco de teste

    fprintf('\nStep 1 - Calculating transformation matrix...\n');

    r = 49; % Número de dimensões na qual se deseja reduzir

    if(strcmp(typeOfClassification, pcaStr))
        transformation_matrix = pca(train_data,r);
    else
        if strcmp(typeOfClassification, nmfStr)
            [w, h] = nmf(train_data, r, 0);
            transformation_matrix = transpose(h);
        else
            end

        end

    end

    fprintf('\nStep 2 - Verifying the test set...\n');

    train_data_transformed = train_data * transformation_matrix;
    test_data_transformed = test_data *transformation_matrix;

    train_set_size = size(train_data_transformed,1);
    test_set_size = size(test_data_transformed,1);
```

```

classification_labels = zeros(test_set_size,1);

fprintf('\nStep 3 - Calculating the distances...\n');

for i=1:test_set_size

    distances = zeros(1,train_set_size);
    test_sample = test_data_transformed(i,:);

    for j=1:train_set_size
        train_sample = train_data_transformed(j,:);
        distances(j) = eucl_distance(test_sample,train_sample);
    end

    [min_distance min_index] = min(distances);
    classification_labels(i) = train_labels(min_index);

end

fprintf('\nStep 4 - Showing the result...\n');

accuracy_vector = (test_labels==classification_labels);
accuracy = (sum(accuracy_vector)/test_set_size);

fprintf('\nUsing %s with r = 49', typeOfClassification);
fprintf('\nAccuracy: %.2f%% \n',accuracy * 100);
fprintf('\n');
end

% Distancia euclidiana
function dist = eucl_distance(a, b)
    dim=size(a,2);
    dist=0;

    for i=1:dim
        dist=power(a(i)-b(i),2)+dist;
    end

    dist=sqrt(dist);
end

% algoritmo NMF [Lee]
function [w,h] = nmf (v,r)

% testa valores negativos em V
if min(min(v)) < 0
    error('matrix entries can not be negative');
    return
end
if min(sum(v,2)) == 0

```

```
        error('not all entries in a row can be zero');
    return
end

[n,m]=size(v);
stopconv=40;      % critério de parada
niter = 2000;     % numero maximo de iterações

cons=zeros(m,m);
consold=cons;
inc=0;
j=0;

%
% inicializa randomicamente w e h
%
w=rand(n,r);
h=rand(r,m);

for i=1:niter

    % atualização dos valores da nmf

    x1=repmat(sum(w,1)',1,m);
    h=h.*(w'*(v./(w*h)))./x1;
    x2=repmat(sum(h,2)',n,1);
    w=w.*((v./(w*h))*h')./x2;

    % testando convergência

    if(mod(i,10)==0)
        j=j+1;

        h=max(h,eps);w=max(w,eps);

        [y,index]=max(h,[],1);
        mat1=repmat(index,m,1);
        mat2=repmat(index',1,m);
        cons=mat1==mat2;

        if(sum(sum(cons~=consold))==0)
            inc=inc+1;
        else
            inc=0;
        end

        if(inc>stopconv)
            break,
        end

        consold=cons;
    end
end
```

```
end

end

% algoritmo PCA [Smith & Pearson]
function T = pca(X, r)
    S = dispersion(X);
    d=size(X,2);
    [evec eval] = eig(S);
    T=zeros(d,r);
    [sorted_val sorted_index] = sort(diag(eval),'descend');
    for i=1:r
        T(:,i)=evec(:,sorted_index(i));
    end

end

function S = dispersion(X)
    d = size(X,2);
    N = size(X,1);

    mu = mean(X);

    S = zeros(d);

    for i=1:N
        diff = X(i,:)-mu;
        S=S+diff'*diff;
    end

    S = S/(N-1);

end
```