

Universidade Federal do Espírito Santo  
Centro Tecnológico  
Programa de Pós-Graduação em Engenharia Elétrica

Fernando Kentaro Inaba

Representação Esparsa e Modelo de Esparsidade  
Conjunta no Reconhecimento de Faces

Vitória  
2012

**Fernando Kentaro Inaba**

**Representação Esparsa e Modelo de Esparsidade  
Conjunta no Reconhecimento de Faces**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Evandro Ottoni Teatini Salles

Vitória  
2012

Dados Internacionais de Catalogação-na-publicação (CIP)  
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

---

I35r Inaba, Fernando Kentaro, 1986-  
Representação esparsa e modelo de esparsidade conjunta no  
reconhecimento de faces / Fernando Kentaro Inaba. – 2012.  
103 f. : il.

Orientador: Evandro Ottoni Teatini Salles.  
Dissertação (Mestrado em Engenharia Elétrica) – Universidade  
Federal do Espírito Santo, Centro Tecnológico.

1. Reconhecimento de padrões. 2. Processamento de  
imagens. 3. Inteligência artificial. I. Salles, Evandro Ottoni Teatini.  
II. Universidade Federal do Espírito Santo. Centro Tecnológico. III.  
Título.

CDU: 621.3

---

**Fernando Kentaro Inaba**

**Representação Esparsa e Modelo de Esparsidade Conjunta  
no Reconhecimento de Faces**

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica.

Aprovada em 11 de junho de 2012.

**COMISSÃO EXAMINADORA**

---

Prof. Dr. Evandro Ottoni Teatini Salles  
Universidade Federal do Espírito Santo  
Orientador

---

Prof. Dr. Allan de Medeiros Martins  
Universidade Federal do Rio Grande do Norte

---

Prof. Dr. Klaus Fabian Côco  
Universidade Federal do Espírito Santo

*À memória de minha querida avó Margarida.*

## Agradecimentos

Expressar minha gratidão a todos que me apoiaram durante os últimos anos, em alguns parágrafos sem cometer injustiças, me parece impraticável. Portanto, agradeço à todos que contribuíram, de forma direta ou indireta, para que esse trabalho fosse concluído. Todavia, me atrevo a acentuar algumas pessoas e instituições sem as quais esse trabalho seria inviabilizado. Muito obrigado...

Ao meu orientador, professor Evandro Ottoni Teatini Salles, pela paciência, ensinamentos e tempo disposto nas reuniões que moldaram o trabalho. Pela confiança depositada em mim e pela atenção prestada ao longo desses dois anos, principalmente no momento em que mais precisei. Pela introdução ao tema de amostragem compressiva, extremamente interessante, que desencadeou a pesquisa sobre os temas abordados nessa dissertação.

Ao CNPq pelo apoio financeiro concedido e ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da UFES pela oportunidade do mestrado. Aos colegas estudantes da Pós-graduação em Engenharia Elétrica pelos momentos de estudo e distração, em especial aos colegas do laboratório CISNE. Aos amigos e familiares pelo apoio, incentivo e suporte.

Aos professores Moisés Ribeiro, Jussara Fardin e Mário Sarcinelli do programa de Pós-graduação em Engenharia Elétrica pelos conhecimentos transmitidos e pela atenção. Ao professor Thomas Rauber do programa de Pós-graduação em Informática que permitiu a experiência de realizar um trabalho em conjunto, em um tema que me foi útil durante minha pesquisa.

Ao professor Marcio Có, pelo apoio e incentivo para realizar o mestrado. Ao professor Klaus Fabian Côco, pela paciência e conhecimento compartilhado, não apenas nos últimos dois anos, mas também enquanto eu era aluno de graduação. Sempre disposto a ajudar, foi decisivo para que eu firmasse o interesse por pesquisa. Realizou um elo

que resultou em uma importante mudança de rumo em meu mestrado.

Por último, mas de forma alguma menos importante, agradeço aos meus pais. Atribuo todo e qualquer sucesso profissional e pessoal que tenho ou venha a ter à eles. Trabalharam muito para que eu pudesse ter o melhor possível, mesmo quando exigia sacrifícios da parte deles. Pelo amor incondicional e compreensão durante os momentos de falta, meus sinceros agradecimentos.

## Resumo

### *Representação Esparsa e Modelo de Esparsidade Conjunta no Reconhecimento de Faces*

Fernando Kentaro Inaba

**Resumo:** O trabalho desenvolvido nesta dissertação propõe a utilização do modelo de esparsidade conjunta com complemento de matrizes (JSM-MC) para composição da base de treino no contexto de reconhecimento de faces utilizando o classificador baseado em representação esparsa (SRC). O método proposto visa trabalhar com imagens de faces em diferentes condições de iluminação e oclusão na base de teste e treino. Para oclusões nas imagens de teste, um modelo diferenciado é considerado para abordar o problema. Uma etapa de pré-processamento nas imagens de faces é realizada no intuito de reduzir os efeitos das variações de iluminações presentes nas imagens. Um agrupamento das imagens de treino é realizado visando um menor tempo de processamento. Além disso, uma proposta de modificação no algoritmo SRC é feita de forma a explorar a esparsidade dos coeficientes de representação esparsa. Ao final, os resultados são avaliados usando uma base de dados sujeita a variação de iluminação. Oclusões artificiais são inseridas a fim de investigar o desempenho do sistema nessas condições.

**Palavras-chaves:** Representação Esparsa, Modelo de Esparsidade Conjunta, Reconhecimento de Face, Completamento de Matrizes, Minimização da norma- $\ell_1$



## Abstract

### *Sparse Representation and Joint Sparsity Model in Face Recognition*

Fernando Kentaro Inaba

**Abstract:** This work proposes the usage of a Joint Sparsity Model with Matrix Completion (JSM-MC) for the composition of training set in the context of face recognition using the Sparse Representation-based Classifier (SRC). The proposed work aims to deal with face images in different illumination conditions and occlusions in the test and training set. For occlusions in the test set, an extended version of the algorithm is done to take into account occlusions in the optimization model. A pre-processing step is performed in the face images to reduce the effects of illumination change. A clustering of training images is done to reduce the processing time and a modification in the SRC algorithm is done to explore the sparsity of the sparse representation coefficients. The results are evaluated using a database with different illumination conditions. Artificial occlusions are inserted in the face images to investigate the behavior of the system in those conditions.

**Keywords:** Sparse Representation, Joint Sparsity Model, Face Recognition, Matrix Completion,  $\ell_1$ -minimization

# Lista de Figuras

|     |  |    |
|-----|--|----|
| 1.1 | Exemplos de problemas enfrentados em reconhecimento de faces: (a) variação de expressão [38], (b) pose (CMU <i>Face Database</i> [23]), (c) iluminação [25, 33], (d) oclusão (CMU <i>Face Database</i> [23]) e (e) <i>pixels</i> corrompidos (modificado de [25, 33]). . . . . | 19 |
| 2.1 | (a) Sinal 10-esparso no tempo, (b) Sinal no domínio da frequência e as amostras coletadas, (c) reconstrução pela minimização $\ell_2$ , (d) reconstrução pela minimização $\ell_1$ . . . . .   | 27 |
| 2.2 | (a) Imagem de teste original ( <i>phantom</i> ), (b) domínio de amostragem (Fourier), (c) reconstrução usando <i>backprojection</i> , (d) reconstrução utilizando <i>total variation minimization</i> . . . . .  | 28 |
| 2.3 | Classificação baseado em representação esparsa . . . . .   | 30 |
| 3.1 | Esquema da composição da base de treino . . . . .  | 41 |
| 3.2 | Esquema de classificação . . . . .   | 43 |
| 3.3 | Exemplos de três imagens de face. . . . .  | 44 |
| 3.4 | Gráfico das três imagens vetorizadas. . . . .  | 44 |
| 3.5 | (a) Gráfico da Figura 3.3a vetorizada e (b) gráfico da Figura 3.3c vetorizada. . . . .   | 45 |
| 4.1 | Resultado da equalização de histograma para quatro imagens de face. As imagens superiores são as faces originais e as inferiores são os resultados após o pré-processamento . . . . .  | 49 |
| 4.2 | Imagem comum obtida para o conjunto de faces utilizadas. . . . .   | 50 |

|      |  |    |
|------|--|----|
| 4.3  | Imagens correspondentes às matrizes de posto pequeno e esparsas oriundas da decomposição usando o método JSM-MC. As imagens superiores são algumas das imagens usadas (após a equalização de histograma). As imagens localizadas no meio são referentes as matrizes de posto pequeno e as imagens inferiores correspondem às matrizes esparsas. . . . .  | 51 |
| 4.4  | Comparação entre as imagens originais (superior), com histograma equalizado (meio) e reconstruídas sem a influência das matrizes esparsas (inferior). . . . .  | 52 |
| 4.5  | Sammon plot para visualização do agrupamento realizado utilizando o <i>k-means</i> . Cada agrupamento ( <i>cluster</i> ) é substituído pelo centróide. . .   | 53 |
| 4.6  | <i>Cluster</i> 1. Na imagem da esquerda está o centróide e as outras imagens correspondem ao <i>cluster</i> 1. . . . .   | 53 |
| 4.7  | <i>Cluster</i> 3. Na imagem da esquerda está o centróide e as três imagens correspondem ao <i>cluster</i> 3. . . . .   | 54 |
| 4.8  | <i>Cluster</i> 10. Na imagem da esquerda está o centróide e as demais imagens correspondem ao <i>cluster</i> 10. . . . .   | 54 |
| 4.9  | <i>Cluster</i> 8. Na imagem da esquerda está o centróide e as demais imagens correspondem ao <i>cluster</i> 8. . . . .   | 54 |
| 4.10 | Imagem de teste utilizada para verificar o efeito da oclusão em blocos na estimação do erro e do vetor de pesos . . . . .  | 55 |
| 4.11 | Efeito da oclusão aleatória na estimação do erro e do vetor de pesos. A primeira coluna referente a 10% de ruído aleatório e a última 75% de ruído (25% e 50% de ruído para a segunda e terceira coluna respectivamente). A primeira linha são as imagens com o ruído, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste . . . . . | 56 |

- 4.12 Efeito da oclusão em blocos em posição aleatória, usando um bloco com todos os elementos iguais a zero, na estimação do erro e do vetor de pesos. A primeira coluna referente a 10% de ruído aleatório e a última 75% de ruído (25% e 50% de ruído para a segunda e terceira coluna respectivamente). A primeira linha são as imagens com o ruído após a equalização de histograma, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste . . . . . 57
- 4.13 Efeito da oclusão em blocos em posição aleatória e usando uma imagem não face na estimação do erro e do vetor de pesos. A primeira coluna referente a 10% de ruído aleatório e a última 75% de ruído (25% e 50% de ruído para a segunda e terceira coluna respectivamente). A primeira linha são as imagens com o ruído, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste . . . . . 58
- 4.14 Efeito da oclusão em blocos na estimação do erro e do vetor de pesos. A primeira coluna referente a 0% de ruído dentro do bloco e a última 100% de ruído dentro do bloco (incremento de 25% de ruído no bloco por coluna). A primeira linha são as imagens com o ruído, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste. . . . . 60
- 4.15 Efeito da oclusão em blocos na estimação do erro e do vetor de pesos (sem a etapa de equalização de histograma). A primeira coluna referente a 0% de ruído dentro do bloco e a última 100% de ruído dentro do bloco (incremento de 25% de ruído no bloco por coluna). A primeira linha são as imagens com o ruído, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste . . . . . 62

|      |  |    |
|------|--|----|
| 5.1  | Exemplo das oclusões realizadas nas imagens de teste. . . . .  | 66 |
| 5.2  | Exemplos de imagens de face com <i>pixels</i> aleatórios com níveis de cinza corrompidos. Taxa de oclusão: (a) 25%, (b) 50% e (c) 75%. . . . .   | 68 |
| 5.3  | Exemplos de imagens de face com oclusão em blocos, em posição aleatória, utilizando imagem não relacionada. Taxa de oclusão: (a) 10%, (b) 25% e (c) 50%. . . . .                                 | 71 |
| 5.4  | Exemplos de imagens de treino da mesma classe . . . . .  | 74 |
| 5.5  | Exemplos de imagens de treino da mesma classe após equalização de histograma para imagens (a) sem oclusão, (b) com 10%, (c) 25% e (d) 50% de oclusão aleatória. . . . .                          | 75 |
| 5.6  | Imagem comum da decomposição JSM-MC (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão. . . . .  | 76 |
| 5.7  | Imagens referentes às 4 matrizes de posto pequeno da decomposição JSM-MC (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão. . . . .   | 77 |
| 5.8  | Imagens referentes às 4 matrizes esparsas da decomposição JSM-MC (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão. . . . .   | 78 |
| 5.9  | Imagens reconstruídas utilizando apenas a matriz comum e as matrizes de posto pequeno para as imagens de treino (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão aleatória. . . . .        | 79 |
| 5.10 | Imagens referentes ao agrupamento realizado nas matrizes de posto pequeno da decomposição JSM-MC das imagens de treino (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão aleatória. . . . . | 81 |
| 5.11 | Matriz de Confusão para rejeição de faces sem oclusão variando $\tau$ . . . . .  | 86 |
| 5.12 | Matriz de Confusão para rejeição de faces com oclusão de 25% e variação de $\tau$ . . . . .  | 87 |
| 5.13 | Matriz de Confusão para rejeição de faces com oclusão de 50% e variação de $\tau$ . . . . .  | 88 |
| 5.14 | Curva ROC para rejeição de faces com (a) 0%, (b) 25% e (c) 50% de oclusão. . . . .   | 89 |

# Lista de Tabelas

|      |  |    |
|------|--|----|
| 5.1  | Taxas de acerto obtidas usando o método descrito . . . . .   | 64 |
| 5.2  | Resultados na taxa de reconhecimento sem oclusão. Retirado de [56]. . . . .  | 65 |
| 5.3  | Resultados para oclusão em bloco de 20% e $d = 300$ . . . . .  | 66 |
| 5.4  | Oclusão Aleatória . . . . .  | 69 |
| 5.5  | Resultados na taxa de reconhecimento com oclusões aleatórias nas imagens de teste. Retirado de [54]. . . . .   | 70 |
| 5.6  | Oclusão em blocos (usando imagem não correlacionada) em posição aleatória para diferentes taxas de oclusão . . . . .   | 71 |
| 5.7  | Resultados na taxa de reconhecimento com oclusões em blocos utilizando imagens não correlacionadas. Retirado de [54]. . . . .                                      | 72 |
| 5.8  | Oclusão em blocos (usando imagem não correlacionada) em posição aleatória para diferentes taxas de oclusão usando 10 <i>clusters</i> por classe . . . . .          | 73 |
| 5.9  | PSNR das componentes comum e de posto pequeno para imagens de treino com oclusão aleatória das 4 imagens mostradas na Figura 5.4 . . . . .                         | 74 |
| 5.10 | PSNRs das quatro imagens reconstruídas com e sem a matriz esparsa $E$ para 10%, 25% e 50% de oclusão . . . . .   | 76 |
| 5.11 | PSNRs dos centróides resultantes do agrupamento das componentes de posto pequeno, para imagens de treino com oclusão aleatória, mostrados na Figura 5.10 . . . . . | 80 |
| 5.12 | Testes utilizando oclusões aleatórias de 10%, 25% e 50% na base de treino  | 82 |
| 5.13 | Matriz de Confusão para rejeição de faces sem oclusão. $\tau = 0,2269$ . . . . .   | 84 |
| 5.14 | Matriz de Confusão para rejeição de faces com 25% de Oclusão. $\tau = 0,2405$  | 84 |

5.15 Matriz de Confusão para rejeição de faces com 50% de oclusão.  $\tau = 0,2356$  84

# Sumário

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>  | <b>18</b> |
| 1.1      | Objetivo . . . . .   | 21        |
| 1.2      | Estrutura da dissertação . . . . .                                 | 21        |
| 1.3      | Notações utilizadas ao longo do texto . . . . .                    | 22        |
| 1.4      | Normas . . . . .   | 23        |
| 1.5      | PSNR . . . . .   | 23        |
| 1.6      | Nota computacional e base de dados . . . . .                       | 24        |
| <b>2</b> | <b>Esparsidade aplicada ao reconhecimento de faces</b>             | <b>25</b> |
| 2.1      | Esparsidade . . . . .  | 25        |
| 2.2      | Amostragem Compressiva . . . . .                                   | 26        |
| 2.3      | Classificação baseado em representação esparsa . . . . .           | 29        |
| 2.3.1    | Amostra de teste como combinação linear das amostras de treino     | 30        |
| 2.3.2    | O algoritmo de classificação . . . . .                             | 32        |
| 2.3.3    | Espaço de características de baixa dimensão . . . . .              | 33        |
| 2.3.4    | A versão estendida . . . . .                                       | 33        |
| 2.4      | Representação de conjunto ( <i>ensemble</i> ) de imagens . . . . . | 34        |
| 2.4.1    | Completamento de matrizes . . . . .                                | 35        |
| 2.4.2    | Modelo de esparsidade conjunta . . . . .                           | 35        |
| 2.4.3    | Modelo de esparsidade conjunta com completamento de matrizes       | 36        |



|          |  |           |
|----------|--|-----------|
| <b>3</b> | <b>O método proposto</b>   | <b>40</b> |
| 3.1      | Composição da base de treino . . . . .                                       | 40        |
| 3.2      | O algoritmo de classificação . . . . .                                       | 42        |
| 3.2.1    | Versão estendida . . . . .   | 45        |
| <b>4</b> | <b>Testes</b>  | <b>48</b> |
| 4.1      | Equalização de histograma . . . . .  | 48        |
| 4.2      | Decomposição usando JSM-MC . . . . .   | 49        |
| 4.3      | Agrupamento ( <i>clustering</i> ) para redução da base de treino . . . . .   | 50        |
| 4.4      | Tipos de oclusões . . . . .  | 53        |
| 4.4.1    | Oclusão aleatória . . . . .  | 54        |
| 4.4.2    | Oclusão em blocos . . . . .  | 55        |
| 4.4.3    | Oclusão em blocos com ruído no bloco . . . . .                               | 59        |
| 4.4.4    | Oclusão em blocos sem equalização de histograma . . . . .                    | 59        |
| <b>5</b> | <b>Resultados</b>  | <b>63</b> |
| 5.1      | Imagens de teste sem oclusões . . . . .                                      | 63        |
| 5.2      | Imagens de teste com oclusão . . . . .                                       | 65        |
| 5.3      | Imagens de teste com oclusão utilizando a versão estendida . . . . .         | 68        |
| 5.3.1    | Oclusão aleatória . . . . .  | 68        |
| 5.3.2    | Oclusão em blocos em posição aleatória e imagem não correlacionada . . . . . | 70        |
| 5.4      | Oclusão na base de treino . . . . .  | 72        |
| 5.5      | Rejeição de imagens de teste . . . . .                                       | 83        |
| <b>6</b> | <b>Conclusões e trabalhos futuros</b>  | <b>90</b> |
|          | <b>Referências Bibliográficas</b>  | <b>93</b> |

|   |  |     |
|---|--|-----|
| A | Equalização de Histograma                | 99  |
| B | <i>K-means clustering</i>                | 100 |
| C | Decomposição em Valores Singulares (SVD) | 101 |
| D | <i>Robust PCA</i>                        | 102 |

# Capítulo 1

## Introdução

O reconhecimento de faces é uma tarefa que os humanos executam constantemente e sem esforço no dia a dia [34]. Implementar esta tarefa em um computador é um problema clássico na comunidade de visão computacional. Os interesses pelo desenvolvimento de pesquisas neste problema envolvem duas razões principais. Primeiro, em reconhecimento de face encontra-se muitas das variabilidades comuns encontradas no campo de sistemas de visão geral: iluminação, oclusão, pose, e desalinhamento. Segundo, o reconhecimento de face possui uma grande gama de aplicabilidade. De fato, a construção de um sistema de reconhecimento de face confiável teria uma ampla implicação em verificação de identidade, controle de acesso e segurança pública. Além disso, a recente proliferação de imagens *online* e vídeos acarretaram em novas aplicações como busca de imagens e marcação de fotos (e.g. Google Piccasa e Apple FaceTime)[24].

Uma das primeiras referências de um desenvolvimento de sistema automático de reconhecimento de faces foi na tese de Ph.D. de Kanade [31] em 1973. Até então, a maioria das pesquisas nesta área eram sistemas semi-automático, necessitando a intervenção humana em algum momento (especialmente na extração de características). Após um período dormente, Sirovich e Kirby [49] utilizaram análise de componentes principais (PCA) para representação de faces em baixa dimensão, que posteriormente foi utilizado no trabalho de Turk e Pentland [51] para classificação, revigorando pesquisas em reconhecimento de faces. Outros marcos importantes em reconhecimento de faces incluem [34]: o método *Fisherface* [3, 21], que aplica análise de discriminantes lineares (LDA) após a etapa de PCA para obter maior precisão; a utilização de filtros locais tais como Gabor jets [32, 53] para obtenção de uma extração de características mais efetiva; e o projeto de aprendizado AdaBoost baseado em arquitetura de classificação em cascata para detecção de face em tempo real [52].

As técnicas de reconhecimento de face avançaram significativamente desde o tempo quando o método *Eigenface* por Turk e Pentland [51] foi proposto. Em situações controladas, por exemplo, onde iluminação, oclusão e variação de expressão podem ser controladas, reconhecimento automático de faces pode superar a proeza humana de reconhecimento de face, especialmente quando a base de dado contiver um grande número de faces [34]. Entretanto, apesar de algumas décadas de trabalho nesta área, o reconhecimento de face automático de alta qualidade ainda é um problema desafiador, principalmente em ambientes não controlados onde as imagens de faces sofrem com variação de iluminação, dados corrompidos, oclusão e desalinhamento. Algumas dessas variáveis são ilustradas na Figura 1.1.

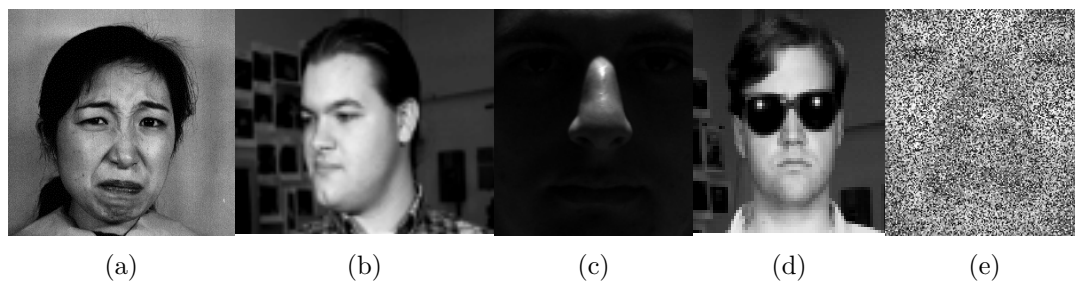


Figura 1.1: Exemplos de problemas enfrentados em reconhecimento de faces: (a) variação de expressão [38], (b) pose (CMU *Face Database* [23]), (c) iluminação [25, 33], (d) oclusão (CMU *Face Database* [23]) e (e) *pixels* corrompidos (modificado de [25, 33]).

Na última década, diversos métodos para tratar uma única variabilidade, tais como iluminação ou oclusão, foram propostos e examinados, mas pouco tem-se feito em tratar múltiplos modos de variação [59]. Em outras palavras, embora um método trate com sucesso um tipo de variação, rapidamente falha quando uma quantidade moderada de outras variações são introduzidas nas imagens de face [24].

Recentemente, a teoria de representação esparsa e amostragem compressiva trouxe uma nova luz para este problema desafiador. De fato, há uma noção muito natural de esparsidade no problema de reconhecimento de face: dada uma imagem de teste, tenta-se encontrar apenas um objeto (uma imagem) entre uma grande quantidade de imagens de diferentes classes (pessoas) que melhor explica tal imagem de teste. Ferramentas relacionadas ao estudo de amostragem compressiva, especialmente minimização da norma- $\ell_1$  e projeções aleatórias, têm inspirado novos algoritmos para reconhecimento de face [24].

Em muitas aplicações é desejável a representação de sinais/imagens usando estruturas elementares escolhidas em um conjunto chamado de dicionário, na esperança de

obter uma representação onde seus coeficientes revelem as informações relevantes. Um exemplo disso é compressão de imagens e áudio. Muitas técnicas de compressão com perdas tais como JPEG, JPEG-2000 e MP3 baseiam-se na observação de que sinais de áudio e imagens digitais possuem uma representação dita esparsa, onde grande parte dos elementos dessa representação são nulos, em termos de uma base apropriada. De maneira simplificada, a compressão é realizada simplesmente guardando somente os maiores coeficientes. Mas escolher os elementos corretos dentro de dicionários é difícil e requer algoritmos complexos [39].

Nesse contexto, Mallat e Zhang [40] introduziram o algoritmo *matching pursuit* que decompõe o sinal em expansões lineares de formas de onda que são relacionados a partir de um dicionário redundante de funções. Tais formas de onda são escolhidas de modo a melhor representar (*match*) as estruturas do sinal. Contudo, a implementação de tal algoritmo é dada por algoritmo guloso (*greedy*).

Para evitar o uso de algoritmos gulosos, que nem sempre conduzem a solução ótima global, Chen e Donoho [15] propuseram um método denominado *basis pursuit* onde um critério global é utilizado de forma a forçar a esparsidade dos coeficientes da representação do sinal em um dicionário super-completo<sup>1</sup> utilizando a norma- $\ell_1$ .

Apesar do objetivo inicial de tais algoritmos não terem sido para classificação, e sim para representação e compressão, Wright et al. [54] utilizaram a representação esparsa de um sinal em respeito a um dicionário super-completo para classificação de faces. Nesse caso, os elementos são as faces de treino, para realizar classificação com o método denominado classificação baseado em representação esparsa (SRC - *sparse representation-based classification*).

Ao invés de utilizar esparsidade para identificar um modelo ou as características relevantes para posteriormente serem usadas para classificação de todas as imagens de teste, a representação esparsa de cada imagem de teste é utilizada diretamente para classificação. Desta forma, seleciona-se de maneira adaptativa as amostras de treino que produzem a representação mais compacta [54].

Ainda motivados pela busca por uma representação onde as informações relevantes são evidenciadas, é possível citar os modelos que realizam a decomposição de um conjunto de sinais (e.g. conjunto de imagens de face vetorizadas de um determinado individuo) em outros sinais que atendam certas restrições. Como exemplo, tem-se o modelo de esparsidade conjunta [2], onde um sinal em um conjunto pode ser visto como uma combinação de uma componente comum que é compartilhada sobre todos

---

<sup>1</sup>Pode ser obtido através da combinação de diferentes dicionários (e.g dicionários Wavelets & Fourier & Gabor)[15].

os sinais do conjunto e uma componente de inovação que é única para este sinal. Outro exemplo é a análise de componentes principais robusta (RPCA - *robust principal component analysis*), proposta por Candès et al. [10], onde uma matriz é decomposta em duas outras matrizes sendo uma de posto pequeno e a outra esparsa. RPCA baseia-se na teoria de completamento de matrizes, onde uma matriz de dados de posto pequeno pode ser recuperada a partir de uma amostragem de poucos elementos desta matriz, pela minimização da norma nuclear, que será definida oportunamente. Zhang e Li [58] utilizam as duas abordagens para compor o modelo de esparsidade conjunta com completamento de matrizes (JSM-MC), onde cada imagem do conjunto é decomposta em uma imagem comum (i.e. a informação compartilhada por todo o conjunto), uma componente com posto pequeno e outra esparsa. Muitos dos métodos citados até o momento utilizam como aplicação o reconhecimento de faces.

## 1.1 Objetivo

Nas conclusões do artigo, Zhang e Li [58] ressaltam a possível utilização do método JSM-MC em reconhecimento de faces, contudo um algoritmo não foi proposto para tal tarefa. O objetivo deste trabalho é elaborar uma metodologia para reconhecimento de faces com diferentes oclusões e variações de iluminação.

Para isso, propõe-se o uso do JSM-MC para composição da base de treino e a utilização da SRC modificado para a classificação das faces de teste, além da utilização de uma etapa de pré-processamento para diminuir os efeitos das variações de iluminação nas imagens de faces.

Para testar o algoritmo proposto, utiliza-se a base de dados Yale B estendida [25] que possui variações de iluminação. Além disso, oclusões artificiais são inseridas na base no intuito de verificar o comportamento do sistema quando a imagem de teste possui oclusões.

## 1.2 Estrutura da dissertação

O trabalho está organizado como segue:

Até o momento, o Capítulo 1 contextualizou o problema de reconhecimento de faces com abordagem esparsa, definindo o objetivo do trabalho. Além disso, será apresentado

a notação utilizada ao longo do texto além de revisar alguns conceitos importantes para o entendimento do texto.

O Capítulo 2 abordará os principais métodos utilizados como base para a proposta elaborada. Especificamente, uma revisão de esparsidade, modelo de esparsidade conjunta, completamento de matrizes, classificação baseada em representação esparsa e modelo de esparsidade conjunta com completamento de matrizes será feita neste capítulo.

O método proposto para reconhecimento de faces é descrito no Capítulo 3. Isto é, a abordagem utilizada para composição da base de treino e o algoritmo de classificação baseado em representação esparsa com uma modificação no critério de atribuição da classe para a imagem testada.

No Capítulo 4, testes serão apresentados utilizando a metodologia descrita no Capítulo 3. Cada etapa da composição da base de treino é testada separadamente e ao final é avaliado o efeito de oclusões nas imagens de face. No capítulo seguinte, os resultados utilizando a abordagem proposta são apresentados, fazendo uma breve comparação com os resultados apresentados por Wright et al. [54]. Além disso, explora-se o potencial do método quando há oclusões na base de treino.

Por fim, o Capítulo 6 apresentará as discussões gerais sobre o material apresentado nesta dissertação, as vantagens e desvantagens do método bem como sugestões para trabalhos futuros.

### 1.3 Notações utilizadas ao longo do texto

Vetores serão representados em letras minúsculas e em negrito. Matrizes serão representadas por letras maiúsculas. O símbolo  $\mathbf{0}$  representa um vetor onde todos os elementos são iguais a zero.  $\langle A, B \rangle$  representa o produto interno euclidiano entre duas matrizes definido por  $\langle A, B \rangle := \text{tr}(A^T B)$  e  $\text{tr}(\cdot)$  é o traço de uma matriz. Aplicado ao contexto de reconhecimento de faces, considera-se a  $j$ -ésima imagem, de dimensão  $w \times h$ , da classe  $i$  (composta por  $n_i$  imagens) como  $X_{i,j} \in \mathbb{R}^{w \times h}$  e em sua forma vetorizada como  $\mathbf{v}_{i,j} \in \mathbb{R}^m$ , onde  $m = wh$ . O conjunto de imagens de treino da  $i$ -ésima classe é representado por  $\mathcal{X}_i = \{X_{i,1}, \dots, X_{i,n_i}\}$  ou em sua forma vetorizada (cada coluna de  $A_i$  corresponde a uma imagem vetorizada) é representada como  $A_i = [\mathbf{v}_{i,1} \ \mathbf{v}_{i,2} \ \dots \ \mathbf{v}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$ .

## 1.4 Normas

Norma de vetores (apesar de ser possível aplicar a matrizes, nesta seção considera-se apenas vetores) é uma generalização de tamanho ou magnitude. Qualquer função real de  $\mathbf{x}$ , denotado por  $\|\mathbf{x}\|$ , pode ser definida como norma se atender as seguintes propriedades [14]:

- 1)  $\|\mathbf{x}\| \geq 0$  para todo  $\mathbf{x}$  e  $\|\mathbf{x}\| = 0$  se e somente se  $\mathbf{x} = \mathbf{0}$
- 2)  $\|\alpha\mathbf{x}\| = |\alpha| \|\mathbf{x}\|$ , para qualquer  $\alpha$  real
- 3)  $\|\mathbf{x}_1 + \mathbf{x}_2\| \leq \|\mathbf{x}_1\| + \|\mathbf{x}_2\|$  para todo  $\mathbf{x}_1$  e  $\mathbf{x}_2$  (desigualdade triangular)

Seja  $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^\top$ . Então alguns exemplos da norma de  $\mathbf{x}$  são

$$\begin{aligned}\|\mathbf{x}\|_1 &:= \sum_{i=1}^n |x_i| \\ \|\mathbf{x}\|_2 &:= \sqrt{\mathbf{x}^\top \mathbf{x}} = \left( \sum_{i=1}^n |x_i|^2 \right)^{1/2} \\ \|\mathbf{x}\|_\infty &:= \max_i |x_i|\end{aligned}$$

Estas são chamadas, respectivamente, norma- $\ell_1$ , norma- $\ell_2$  ou norma euclidiana, e norma infinita.

De maneira geral, define-se norma- $\ell_p$  como

$$\|\mathbf{x}\|_p := \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} \quad (1.1)$$

com  $0 < p \leq \infty$ .

## 1.5 PSNR

Para avaliar a qualidade de uma imagem processada em relação à uma imagem de referência, é comum na literatura utilizar o *RMSE* (*root-mean-squared error*) e o *PSNR* (*peak signal-to-noise ratio*) como métricas de erro. Seja a imagem de referência de tamanho  $M \times N$  denotada por  $X$  e a imagem processada denotada por  $\bar{X}$ . O *RMSE* é dado por [46]

$$RMSE = \sqrt{\frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N [X(i, j) - \bar{X}(i, j)]^2} \quad (1.2)$$



e representa o desvio padrão do erro. O  $PSNR$  entre as imagens é definido classicamente como [46]

$$PSNR = 10 \log_{10} \left( \frac{255}{RMSE} \right) \quad (1.3)$$

para uma imagem em nível de cinzas e oito bits por nível de cinza.

## 1.6 Nota computacional e base de dados

Os algoritmos implementados neste trabalho foram realizados em MATLAB<sup>®</sup> rodando em uma máquina com processador quad-core, 2.4 GHz e 4 GB de memória ram. O algoritmo usado para solução dos problemas de otimização foi o CVX, um pacote para especificar e resolver programas convexos [27, 28].

A base de dados utilizada nessa dissertação é a Yale Face Database B estendida [25, 33], versão cortada (*cropped*), com 2414 imagens de 38 indivíduos distintos em diferentes condições de iluminação. Para cada indivíduo tem-se aproximadamente 64 imagens de  $192 \times 168$  *pixels*.

# Capítulo 2

## Esparsidade aplicada ao reconhecimento de faces

Na última década, esparsidade surgiu como um dos conceitos conducentes em uma variedade de aplicações em processamento de sinais (representação, extração de característica, separação de fonte e compressão, para citar algumas das aplicações). Esparsidade tornou-se teoricamente atrativa e uma propriedade prática em muitas áreas da matemática aplicada [50].

Recentemente, pesquisadores de várias áreas têm preconizado o uso de representações de sinais utilizando dicionários supercompletos [50]. Tais representações diferem da representação tradicional pois oferecem um âmbito maior de geração de elementos (chamados de *atoms*). Potencialmente, isto permite maior flexibilidade na representação do sinal e implica mais efetividade em muitas tarefas de processamento de sinais (restauração, separação, compressão e estimação), incluindo reconhecimento de faces.

### 2.1 Esparsidade

Considere um vetor  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} = [\alpha_1 \ \cdots \ \alpha_n]^\top$ . Este sinal é exatamente ou estritamente esparso se a maioria de seus elementos forem iguais a zero, isto é, se o suporte  $\Lambda(\mathbf{x}) = \{1 \leq i \leq n | \alpha_i \neq 0\}$  é de cardinalidade  $k \ll n$  [50]. Portanto, um sinal é dito  $k$ -esparso quando possui no máximo  $k$  elementos não nulos.

Se um sinal não é esparso, este pode possuir uma representação esparsa em um domínio de transformada apropriado [50]. De fato, esquemas de compressão baseiam-se nesta ideia. Como exemplos, é possível citar o caso da JPEG que utiliza a DCT e a

JPEG-2000 que utiliza *wavelets*. Em ambos os casos, utiliza-se destas transformadas para obter uma representação esparsa da imagem, sendo então, possível realizar a compressão da imagem.

De forma mais geral, é possível representar o sinal  $\mathbf{y}$  como combinações lineares de  $n$  formas de ondas elementares, tais que

$$\mathbf{y} = A\mathbf{x} = \sum_{i=1}^n \mathbf{v}_i \alpha_i, \quad (2.1)$$

onde  $\alpha_i$  são os coeficientes de representação de  $\mathbf{y}$  no dicionário  $A = [ \mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_n ]$  (uma matriz  $m \times n$ ). Portanto, se for possível representar  $\mathbf{y}$  usando poucos elementos de  $A$  tem-se uma representação esparsa do sinal  $\mathbf{y}$ . Esta observação é a base de uma teoria denominada *Compressive Sensing*.

## 2.2 Amostragem Compressiva

O interesse em esparsidade surgiu devido a nova teoria de amostragem, *compressive sensing* (também chamado de *compressive sampling*, *compressed sensing* ou amostragem compressiva), que fornece uma alternativa ao bem conhecido teorema de amostragem de Shannon [12, 13, 18]. Amostragem compressiva utiliza o conhecimento a priori de que o sinal é esparso (ou que possui uma representação esparsa em uma base apropriada) enquanto a teoria de Shannon foi projetada para sinais com frequência limitada em banda [50].

Estabelecendo uma relação direta entre amostragem e esparsidade, amostragem compressiva teve um grande impacto em vários campos científicos tais como codificação e teoria da informação, aquisição e processamento de sinais e imagens, imageamento médico e geofísico além de análise de dados astronômicos. Outra contribuição para o sucesso de *compressive sampling* é que problemas inversos tradicionais, como reconstrução de imagem tomográfica, podem ser vistos como problemas de amostragem compressiva [12, 37]. Além disso, *compressive sampling* fornece suportes teóricos fortes pela busca da solução esparsa pois tal solução pode ser (sobre certas condições) a exata. Estas razões explicam o porquê de, em apenas poucos anos após o surgimento dos primeiros trabalhos publicados em amostragem compressiva, centenas de artigos já aparecem nesta área [50].

A Figura 2.1 mostra o potencial de amostragem compressiva, onde é apresentado um exemplo de recuperação de um sinal 10-esparso  $\mathbf{x} \in \mathbb{R}^{300}$  (Figura 2.1 (a)) a partir de apenas 30 amostras (indicadas pelas 30 circunferências na Figura 2.1 (b)). Ao

observar o sinal no domínio da frequência, pode-se pensar que a reconstrução a partir de 30 amostras seria impossível. De fato, o sinal reconstruído utilizando a tradicional minimização  $\ell_2$  (Figura 2.1 (c)) é bem diferente do sinal original. De forma surpreendente, a minimização  $\ell_1$  (Figura 2.1 (d)) retorna a reconstrução exata, isto é, sem erro de reconstrução.

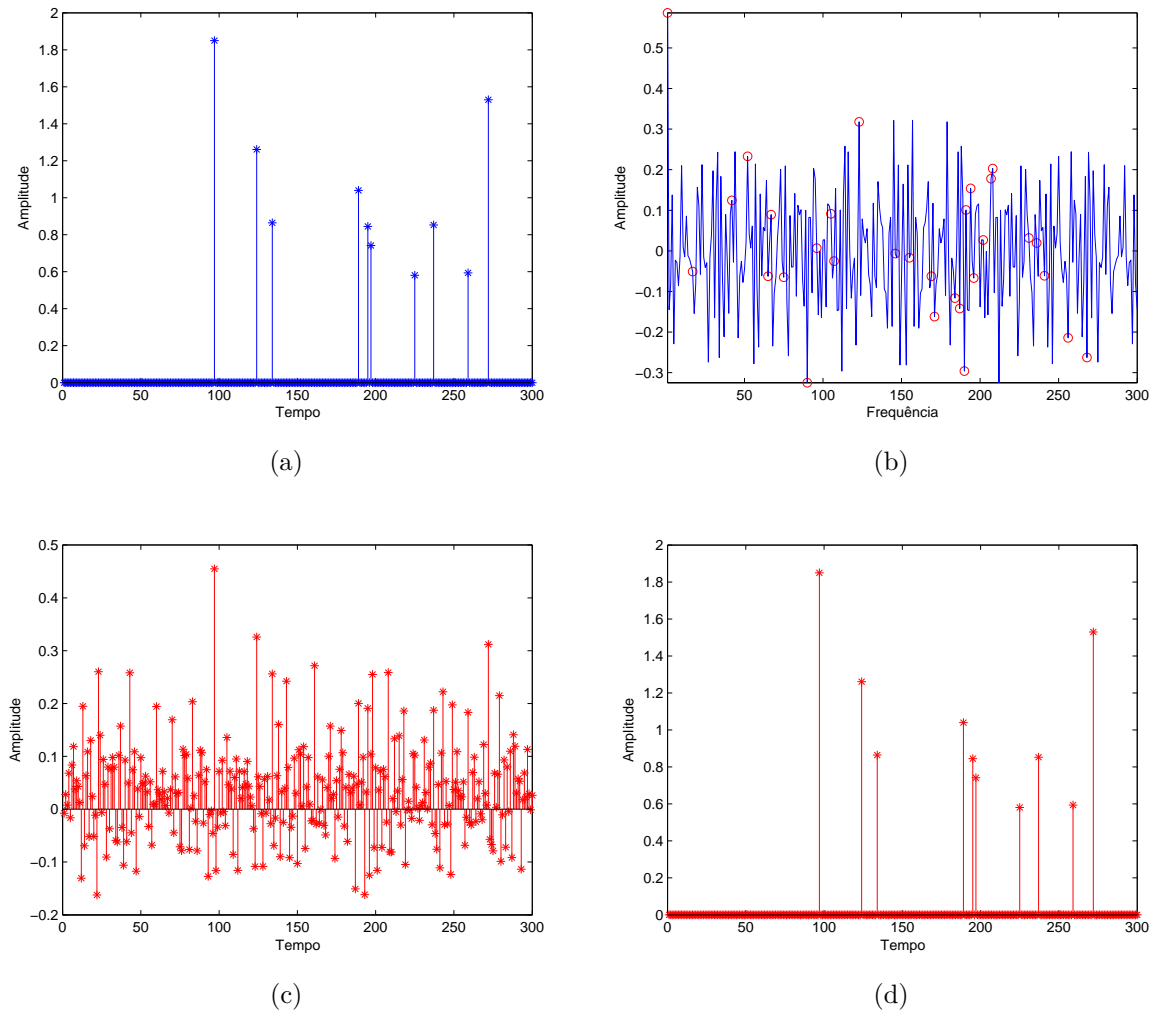


Figura 2.1: (a) Sinal 10-esparso no tempo, (b) Sinal no domínio da frequência e as amostras coletadas, (c) reconstrução pela minimização  $\ell_2$ , (d) reconstrução pela minimização  $\ell_1$

Um segundo exemplo é a imagem de ressonância magnética, representado por uma imagem sintética (*phantom*) mostrado na Figura 2.2 (a). Neste caso, um dispositivo escaneia um paciente através de medidas no domínio da frequência dentro de uma geometria radial. A Figura 2.2 (b) descreve tal conjunto de amostragem da transformada de Fourier 2D. Uma vez que o processo é demorado e desconfortável para o paciente

deseja-se fazer o menor número de medidas [22]. A técnica conhecida como *Total variation minimization* [12], que possui relação com a minimização  $\ell_1$ , é então considerada como método de reconstrução. Afim de comparar, a Figura 2.2 (c) mostra a recuperação pelo algoritmo tradicional *backprojection*. A Figura 2.2 (d), mostra a reconstrução usando *total variation minimization*.

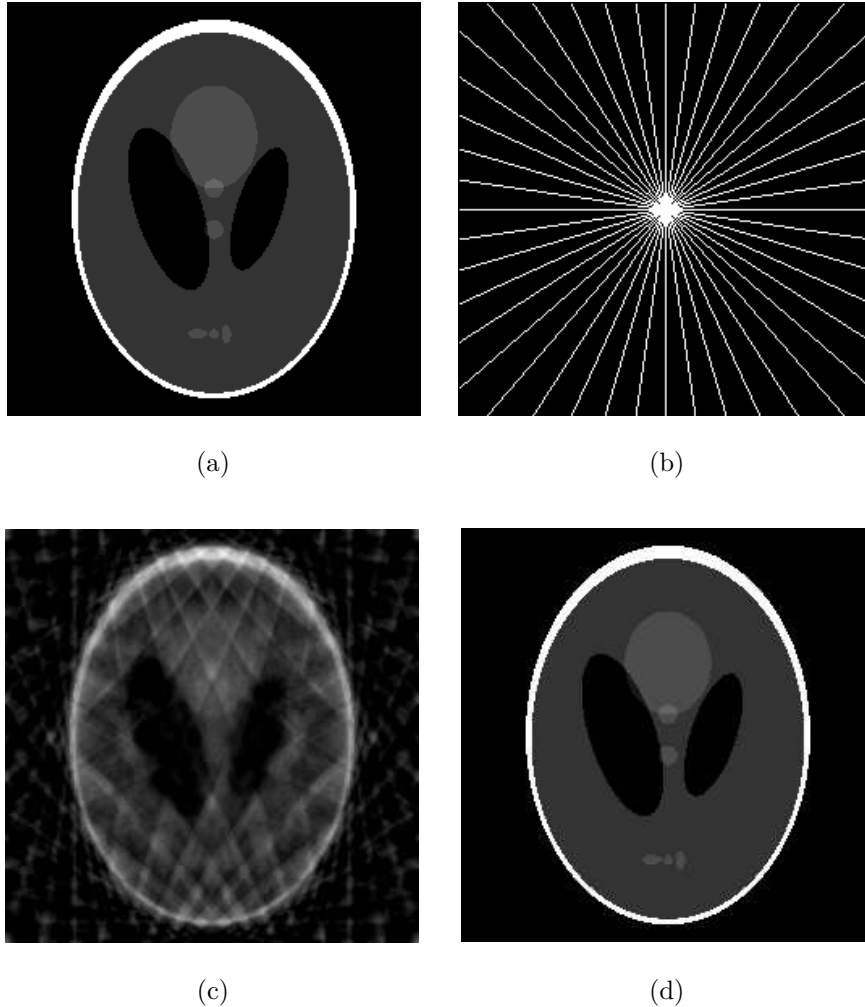


Figura 2.2: (a) Imagem de teste original (*phantom*), (b) domínio de amostragem (Fourier), (c) reconstrução usando *backprojection*, (d) reconstrução utilizando *total variation minimization*

O objetivo inicial destes trabalhos não é inferência ou classificação por si, mas representação e compressão de sinais, possivelmente usando taxas de amostragens menores que o limite Shannon-Nyquist [8]. O desempenho dos algoritmos foram então medidos em termos da esparsidade da representação e fidelidade com os sinais originais. Além disso, os elementos do dicionário não devem, necessariamente, ter algum sentido se-

mântico – são geralmente escolhidos a partir de bases padrões (e.g., Fourier, Wavelet, Curvelet e Gabor), ou mesmo geradas a partir de matrizes aleatórias [13, 8]. Não obstante, a representação mais esparsa é naturalmente discriminativa: dentre todos os subconjuntos de vetores base, seleciona-se o subconjunto que expressa de maneira mais compacta o sinal de entrada e rejeita todas as outras possibilidades.

Neste sentido, Wright et al. [54] explora a natureza discriminativa da representação esparsa para realizar a classificação. Ao invés de usar dicionários genéricos citados anteriormente, representa-se uma amostra de teste em um dicionário supercompacto onde os elementos base são as próprias amostras de treino.

## 2.3 Classificação baseado em representação esparsa

O classificador apresentado por Yang et al. [56], *Sparse Representation-based Classification* (SRC), baseia-se na representação de uma imagem de teste como combinação linear das imagens do conjunto de treino, contudo utilizando o menor número possível de imagens deste conjunto. Em outras palavras, espera-se que uma imagem de teste, de uma determinada classe, possa ser representada apenas pelas imagens de treino pertencentes à mesma classe da imagem de teste. Tal representação é esparsa e pode ser determinada utilizando minimização- $\ell_1$ .

Considere a Figura 2.3. O quadrado,  $\mathbf{y}$ , é o objeto de teste e deseja-se determinar a classe deste objeto a partir do conjunto de treino  $A$ . Para isso, representa-se  $\mathbf{y}$  como combinação linear do conjunto de treino. Claramente,  $\mathbf{y}$  pertence à *classe*<sub>3</sub> e portanto, o vetor de representação  $\hat{\mathbf{x}}_1$ , conforme indicado na figura, que possui elementos diferentes de zero apenas nas posições correspondentes à *classe*<sub>3</sub>, revela a real classe de  $\mathbf{y}$ . Observe que esta não é a única solução ( $\hat{\mathbf{x}}$ ) possível para o problema. A soma de todos os elementos de  $A$  também resulta em um quadrado preenchido. Entretanto, tal solução não é esparsa e determinar a classe real de  $\mathbf{y}$  já não seria mais possível para esta solução.

Nesta seção é apresentado o algoritmo de classificação baseada em representação esparsa introduzido por Yang et al. [56].

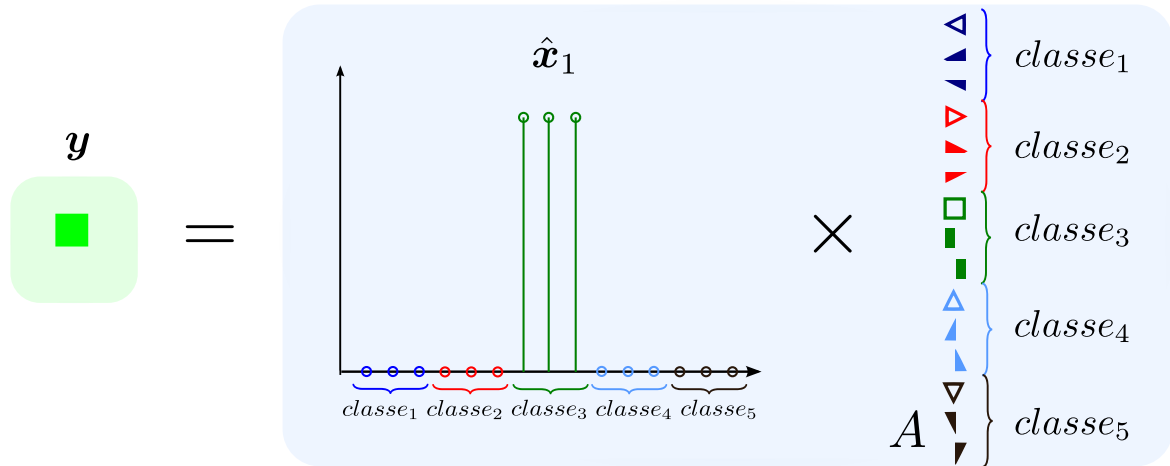


Figura 2.3: Classificação baseado em representação esparsa

### 2.3.1 Amostra de teste como combinação linear das amostras de treino

Seja  $A_i = [ \mathbf{v}_{i,1} \ \mathbf{v}_{i,2} \ \cdots \ \mathbf{v}_{i,n_i} ] \in \mathbb{R}^{m \times n_i}$  amostras de treino da  $i$ -ésima classe, qualquer nova amostra  $\mathbf{y} \in \mathbb{R}^m$  da mesma classe poderá ser aproximadamente representada por

$$\mathbf{y} = \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_i}\mathbf{v}_{i,n_i}, \quad (2.2)$$

com  $\alpha_{i,j} \in \mathbb{R}$ ,  $j = 1, 2, \dots, n_i$ .

Uma vez que a classe da amostra de teste é desconhecida, define-se uma nova matriz  $A$  para todo o conjunto de treino como a concatenação das  $n = n_1 + \cdots + n_k$  amostras de treino de todas as  $k$  classes:

$$\begin{aligned} A &= [ A_1 \ A_2 \ \cdots \ A_k ] \\ &= [ \mathbf{v}_{1,1} \ \mathbf{v}_{1,2} \ \cdots \ \mathbf{v}_{1,n_1} \ \mathbf{v}_{2,1} \ \mathbf{v}_{2,2} \ \cdots \ \mathbf{v}_{2,n_2} \ \cdots \ \mathbf{v}_{k,1} \ \mathbf{v}_{k,2} \ \cdots \ \mathbf{v}_{k,n_k} ]. \end{aligned} \quad (2.3)$$

Então a representação linear de  $\mathbf{y}$  pode ser escrita em termos de todas as amostras de treino como

$$\mathbf{y} = A\mathbf{x}_0 \in \mathbb{R}^m, \quad (2.4)$$

onde  $\mathbf{x}_0 = [ 0 \ \cdots \ 0 \ \alpha_{i,1} \ \alpha_{i,2} \ \cdots \ \alpha_{i,n_i} \ 0 \ \cdots \ 0 ]^T \in \mathbb{R}^n$  é o vetor de coeficientes onde os únicos elementos diferentes de zero são os associados a  $i$ -ésima classe.

Como os elementos de  $\mathbf{x}_0$  codificam a identidade da amostra de teste  $\mathbf{y}$ , é natural a tentativa de sua obtenção pela solução do sistema linear de equações  $\mathbf{y} = A\mathbf{x}$ . Se

$m > n$ , o sistema de equações  $\mathbf{y} = A\mathbf{x}$  é sobredeterminado, e geralmente a correta solução,  $\mathbf{x}_0$ , pode ser encontrada. Entretanto, no contexto de reconhecimento de faces, este sistema é tipicamente subdeterminado e portanto a solução não é única. De maneira a contornar isto, é comum a escolha da solução que possui a menor norma- $\ell_2$ .

$$(\ell_2) : \quad \hat{\mathbf{x}}_2 = \arg \min \|\mathbf{x}\|_2 \quad \text{sujeito a} \quad A\mathbf{x} = \mathbf{y} \quad (2.5)$$

Este problema pode ser facilmente resolvido utilizando a pseudo-inversa de  $A$ , contudo a solução  $\hat{\mathbf{x}}_2$  é geralmente densa, com vários elementos diferentes de zero correspondentes às amostras de treinos de diversas classes. Como mencionado anteriormente, espera-se que a solução seja esparsa e por isso a solução do problema de otimização

$$(\ell_0) : \quad \hat{\mathbf{x}}_0 = \arg \min \|\mathbf{x}\|_0 \quad \text{sujeito a} \quad A\mathbf{x} = \mathbf{y}, \quad (2.6)$$

onde  $\|\cdot\|_0$  representa o número de elementos diferentes de zero do vetor é mais apropriada. Contudo, este problema é *NP*-completo [1], tornando computacionalmente inviável no contexto de reconhecimento de faces.

Trabalhos recentes que descrevem a teoria de *compressive sensing* mostram que se a solução  $\mathbf{x}_0$  é suficientemente esparsa, a solução para o problema (2.6) é igual a solução do seguinte problema de otimização:

$$(\ell_1) : \quad \hat{\mathbf{x}}_1 = \arg \min \|\mathbf{x}\|_1 \quad \text{sujeito a} \quad A\mathbf{x} = \mathbf{y}. \quad (2.7)$$

O problema (2.7) pode ser resolvido em tempo polinomial utilizando métodos de programação linear [16, 54].

O modelo (2.4) pode ser modificado, quando a igualdade não é verdadeira (como no caso de presença de ruído), para considerar explicitamente o ruído

$$\mathbf{y} = A\mathbf{x}_0 + \mathbf{z}, \quad (2.8)$$

onde  $\mathbf{z} \in \mathbb{R}^m$  é o ruído com energia limitada  $\|\mathbf{z}\|_2 < \epsilon$ . A solução  $\mathbf{x}_0$  pode ser aproximadamente recuperada resolvendo-se o problema de minimização- $\ell_1$  estável

$$(\ell_{1s}) : \quad \hat{\mathbf{x}}_1 = \arg \min \|\mathbf{x}\|_1 \quad \text{sujeito a} \quad \|A\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon. \quad (2.9)$$

A letra  $s$  de  $\ell_{1s}$  refere-se à estável (do inglês, *stable*). Dificilmente a igualdade  $A\mathbf{x} = \mathbf{y}$  da restrição do problema (2.7) é assegurada, especialmente na presença de ruídos (considere a princípio ruídos de baixa potência, já que para oclusões/ruídos de maior potência uma outra abordagem será apresentada posteriormente). Portanto, na versão estável tolera-se um ruído de energia limitada  $\|\mathbf{z}\|_2 < \epsilon$ , para garantir que a restrição do problema seja atendida e portanto, haja convergência do algoritmo.



### 2.3.2 O algoritmo de classificação

Dado uma nova amostra de teste,  $\mathbf{y}$ , pertencente a uma das classes do conjunto de treino, primeiramente sua representação esparsa,  $\hat{\mathbf{x}}_1$ , é determinada utilizando (2.7) ou (2.9). Idealmente, os elementos diferentes de zero de  $\hat{\mathbf{x}}_1$  são associados com as colunas de  $A$  de uma única classe, e é possível atribuir tal imagem de teste a esta classe. Entretanto, ruídos e erros de modelagem podem acarretar em pequenos elementos diferentes de zero associados a múltiplas classes.

No algoritmo proposto por Yang et al. [56], a amostra de teste é classificada baseada em quão bem os coeficientes associados com todas as amostras de treino de cada classe reconstróem  $\mathbf{y}$ . Isto é, para cada classe  $i$ , seja  $\delta_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$  a função característica que seleciona os coeficientes associados com a  $i$ -ésima classe. Para  $\mathbf{x} \in \mathbb{R}^n$ ,  $\delta_i(\mathbf{x}) \in \mathbb{R}^n$  é um novo vetor onde os únicos elementos que podem assumir valores diferentes de zero são os associados com a classe  $i$ , em outras palavras, todos os elementos não associados à classe  $i$  recebem valor zero. O critério de classificação de  $\mathbf{y}$  é a classe que possui o menor resíduo, i.e.,

$$\arg \min_i r_i(\mathbf{y}) \doteq \| \mathbf{y} - A\delta_i(\hat{\mathbf{x}}_1) \|_2 \quad (2.10)$$

O Algoritmo 1 resume o procedimento de reconhecimento proposto por Yang et al. [56].

---

**Algoritmo 1:** *Sparse Representation-based Classification* (SRC) [56]

---

- 1: **Entrada:** Matriz das amostras de treino  $A = [ A_1 \ \cdots \ A_k ] \in \mathbb{R}^{m \times n}$  para  $k$  classes. Amostra de teste  $\mathbf{y} \in \mathbb{R}^m$ . Tolerância de erro  $\epsilon > 0$ .
- 2: Normalizar as colunas de  $A$  para ter norma  $\ell_2$  unitária.
- 3: Resolver o problema de minimização- $\ell_1$ :

$$\hat{\mathbf{x}}_1 = \arg \min_x \| \mathbf{x} \|_1 \quad \text{sujeito a} \quad A\mathbf{x} = \mathbf{y}.$$

(Ou sua versão estável (2.9))

- 4: Determinar os resíduos  $r_i(\mathbf{y}) = \| \mathbf{y} - A\delta_i(\hat{\mathbf{x}}_1) \|_2$  para  $i = 1, \dots, k$ .
  - 5: **Saída:**  $Classe(\mathbf{y}) = \arg \min_i r_i(\mathbf{y})$ .
-

### 2.3.3 Espaço de características de baixa dimensão

Uma etapa comum em sistemas de classificação é a extração de características. Para o caso específico de interesse deste trabalho, é possível citar *Eigenfaces* [51], *Fisherfaces* [3] e *Laplacianfaces* [29] como métodos comumente encontrados na literatura. Dependendo do classificador escolhido, a escolha das características são consideradas críticas para o sucesso do algoritmo [54].

No trabalho apresentado por Yang et al. [56], *randomfaces* são utilizadas para realizar a extração de características.

**Definição 2.1.** *Randomfaces* [56]: Considere a matriz de transformação  $R \in \mathbb{R}^{d \times m}$  onde os elementos são amostrados de forma independentes a partir de uma distribuição normal com média zero, cada linha é normalizada para ter tamanho unitário. Os vetores linhas de  $R$  podem ser vistos como  $d$  faces aleatórias em  $\mathbb{R}^m$ .

Portanto, a projeção do espaço da imagem para o espaço de característica é representado pela matriz  $R \in \mathbb{R}^{d \times m}$  com  $d \ll m$ . Aplicando  $R$  aos dois lados de (2.4), tem-se

$$\tilde{\mathbf{y}} = R\mathbf{y} = RA\mathbf{x}_0 \in \mathbb{R}^d, \quad (2.11)$$

e a solução pode ser encontrada resolvendo o seguinte problema de minimização- $\ell_1$  reduzido:

$$(\ell_1^r): \quad \hat{\mathbf{x}}_1 = \arg \min \|\mathbf{x}\|_1 \quad \text{sujeito a} \quad \|RA\mathbf{x} - \tilde{\mathbf{y}}\|_2 \leq \epsilon, \quad (2.12)$$

para um dado erro de tolerância  $\epsilon \geq 0$ . Portanto, no Algoritmo 1, a matriz  $A$  de imagens de treino é substituída pela matriz  $RA \in \mathbb{R}^{d \times n}$  de dimensão  $d$ , e a amostra de teste  $\mathbf{y}$  por  $\tilde{\mathbf{y}}$ .

Uma das grandes vantagens da utilização de *randomfaces* é que podem ser facilmente gerados, já que a transformação  $R$  é independente do conjunto de treino. Além disso, os resultados obtidos por Yang et al. [56] indicam que, não importam as características escolhidas, mesmo projeções aleatórias, o desempenho para o Algoritmo 1 deve ser tão bom caso fossem cuidadosamente escolhidas as características.

### 2.3.4 A versão estendida

Em várias situações, onde sistemas de reconhecimento de faces são utilizados, as imagens de teste,  $\mathbf{y}$ , podem apresentar oclusões. Nesses casos, o modelo (2.4) deve ser

modificado de maneira a contemplar a oclusão. Especificamente, o modelo considerado é [54]:

$$\mathbf{y} = \mathbf{y}_0 + \mathbf{e}_0 = A\mathbf{x}_0 + \mathbf{e}_0, \quad (2.13)$$

com  $\mathbf{e}_0$  sendo o vetor de erros/occlusão. Nesse modelo, considera-se que o vetor de erros/occlusão é também esparsa. Logo, os elementos diferentes de zero do vetor  $\mathbf{e}_0$  correspondem aos elementos de  $\mathbf{y}$  que estão oclusos ou corrompidos. Tais elementos podem ter magnitudes arbitrárias e portanto não podem ser ignoradas ou tratadas com as técnicas descritas anteriormente para pequenos ruídos ( $\ell_{1_s}$ ).

O modelo pode ser reescrito como

$$\mathbf{y} = [A \quad I] \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{e}_0 \end{bmatrix} \doteq B\mathbf{w}_0 \quad (2.14)$$

onde  $\mathbf{w}_0 = [\mathbf{x}_0 \quad \mathbf{e}_0]^T$  e  $B = [A \quad I]$ . A abordagem proposta por Wright et al. [54] para encontrar a solução  $\mathbf{w}_0$  é a busca pela solução mais esparsa utilizando o seguinte problema de minimização- $\ell_1$  estendido:

$$(\ell_1^e) : \quad \hat{\mathbf{w}}_1 = \arg \min \|\mathbf{w}\|_1 \quad \text{sujeito a} \quad B\mathbf{w} = \mathbf{y} \quad (2.15)$$

onde  $\mathbf{w} = [\mathbf{x} \quad \mathbf{e}]^T$  e  $B = [A \quad I]$ . Vale ressaltar que a matriz estendida  $B$  utilizada, considera que a oclusão  $\mathbf{e}$  é esparsa, contudo, pode ser modificada para o caso onde  $\mathbf{e}$  é esparsa em outra base,  $A_e$ . Para isso, faz-se  $B = [A \quad A_e]$  ao invés de utilizar a matriz identidade,  $I$ . Nos testes realizados nesta dissertação considera-se que as oclusões já são naturalmente esparsas conforme sugerido em [54].

## 2.4 Representação de conjunto (*ensemble*) de imagens

Na seção anterior, o método SRC foi apresentado. É possível observar que a solução (vide Algoritmo 1) depende fortemente do conjunto de treino. Logo, idealmente as imagens de treino de uma classe devem representar bem a classe em questão para que não seja necessário contribuições de outras classes e assim, obter uma solução esparsa que revele a identidade da imagem de teste. Nesta seção, modelos de representação de conjuntos (*ensemble*) de imagens correlacionadas são apresentados. Em reconhecimento de faces, imagens de uma mesma classe naturalmente formam tal conjunto de imagens correlacionadas. A utilização de um modelo, onde características relevantes fiquem evidenciadas, pode aumentar o desempenho do classificador apresentado na seção anterior.

Em especial o modelo de esparsidade conjunta com completamento de matrizes (JSM-MC) proposta por Zhang e Li [58] é apresentado. Tal modelo é adotado para compor a base de treino conforme descrito na Seção 3.1. Entretanto, dois conceitos são necessários para o entendimento do JSM-MC: completamento de matrizes e modelo de esparsidade conjunta.

### 2.4.1 Completamento de matrizes

Na teoria de completamento de matrizes [11], uma matriz  $L$  de posto pequeno pode ser recuperada a partir de um subconjunto pequeno de seus elementos pela minimização da norma nuclear:

$$\begin{aligned} &\text{minimizar} && \| Y \|_* \\ &\text{sujeito a} && \Omega(Y) = \Omega(L) \end{aligned} \quad (2.16)$$

onde  $\Omega(\cdot)$  é um subconjunto,  $Y$  a matriz de posto pequeno recuperada e  $\| \cdot \|_*$  é a norma nuclear, que é a soma dos valores singulares da matriz. Para solução de (2.16), é possível aplicar o método *Singular Value Thresholding* (SVT) [7]

$$\mathcal{D}_\tau(Y) = US_\tau(\Sigma)V' \quad (2.17)$$

onde  $U$ ,  $\Sigma$  e  $V$  são a decomposição em valores singulares de  $Y$  e  $S_\tau(\cdot)$  corresponde ao operador *soft thresholding*

$$\mathcal{S}_\tau(x) = \begin{cases} x - \tau & \text{se } x - \tau \geq 0, \\ x + \tau & \text{se } x + \tau \leq 0, \\ 0 & \text{caso contrário.} \end{cases} \quad (2.18)$$

Para recuperação exata,  $L$  deve possuir posto pequeno [11].

### 2.4.2 Modelo de esparsidade conjunta

Modelos de esparsidade conjunta são uma generalização do conceito de esparsidade (em uma base apropriada) quando considera-se um conjunto (*ensemble*) de sinais sendo esparsos conjuntamente [2]. Formalmente, dado um conjunto

$$\mathcal{X}_i = \{X_{i,1}, \dots, X_{i,n_i}\} \quad (2.19)$$

de imagens de treino da  $i$ -ésima classe, cada imagem  $X_{i,j}$  pode ser decomposta em duas partes: uma parte comum,  $z_{C_i}$  e outra denominada inovação,  $z_{i,j}$  [58]. Ou seja,

$$X_{i,j} = z_{C_i} + z_{i,j}, \quad (2.20)$$

onde  $j = 1, \dots, n_i$ . Assim, todas as imagens do conjunto  $\mathcal{X}_i$  compartilham a mesma componente comum,  $z_{C_i}$ , e cada imagem  $X_{i,j}$  do conjunto possui uma componente de inovação,  $z_{i,j}$ , que representa a parcela da imagem que não é modelada pela componente comum do conjunto.

Existem diferentes modelos dependendo das considerações feitas acerca da esparsidade da parte comum e inovação. Em especial, utiliza-se o modelo *Joint Sparsity Model 3* (JSM-3) [2], onde a parte comum é arbitrária e a inovação é esparsa, i.e.,  $\|z_{i,j}\|_0 \leq t$ , onde  $t$  é uma constante pequena. Embora isto, a princípio, pareça contraditório a definição dada para modelos de esparsidade conjunta, já que espera-se que o conjunto seja esparso conjuntamente, o modelo é pertinente uma vez que é possível obter uma representação esparsa da parte comum do JSM-3 (que é arbitrária) em respeito à uma base apropriada.

### 2.4.3 Modelo de esparsidade conjunta com completamento de matrizes

Embora as técnicas JSM-3 e completamento de matrizes tenham várias aplicações, algumas dificuldades são encontradas ao aplicá-las em reconhecimento de face [58]. Considere  $\mathcal{X}_i$  como definido em (2.19) em diferentes condições de iluminação/oclusão. Com o modelo JSM-3, a parte de inovação é considerada esparsa. Entretanto, iluminação (ou grandes oclusões) podem afetar toda a imagem, e dificilmente são esparsas em qualquer base. Nas abordagens baseadas em completamento de matrizes (e.g. PCA Robusto) assume-se que a matriz de posto pequeno tenha a parte comum do conjunto de imagens, o que é muito restritivo, uma vez que uma imagem de face natural, limpa (espera-se que a imagem comum seja uma face sem o efeito da iluminação/oclusão) não possui, em geral, posto pequeno.

Para contornar estas dificuldades, Zhang e Li [58] introduziram o modelo *Joint Sparsity Model with Matrix Completion* (JSM-MC). Neste modelo, cada imagem  $X_{i,j}$  do conjunto  $\mathcal{X}_i$  é representada por

$$X_{i,j} = C_i + L_{i,j} + E_{i,j} \quad (2.21)$$

onde  $C_i$  é a parte comum,  $L_{i,j}$  uma matriz de posto pequeno e  $E_{i,j}$  a inovação esparsa. Com este modelo, o problema pode ser formulado da seguinte forma:

$$\begin{aligned} \{C_i, \{L_{i,j}\}, \{E_{i,j}\}\} = & \arg \min_{C_i, \{L_{i,j}\}, \{E_{i,j}\}} \sum_j \|L_{i,j}\|_* + \lambda_{i,j} \|E_{i,j}\|_1 \\ & \text{sujeito a } X_{i,j} = C_i + L_{i,j} + E_{i,j} \end{aligned} \quad (2.22)$$

onde  $\lambda_{i,j}$  é um escalar. A minimização da norma nuclear e da norma  $\ell_1$  forçam, respectivamente, a matriz  $L_{i,j}$  ter posto pequeno e a matriz  $E_{i,j}$  ser esparsa. A restrição de (2.22) assegura que o modelo (2.21) seja obedecido. Assim, dentre todas as decomposições possíveis, busca-se por aquela que minimiza a combinação ponderada da norma nuclear e a norma  $\ell_1$ .

Para solução do problema (2.22), Zhang e Li [58] baseiam-se no método Lagrangiano aumentado. A função de Lagrangiano Aumentado associado a (2.22) é

$$\begin{aligned} \mathcal{L}(C_i, \{L_{i,j}\}, \{E_{i,j}\}, \{\Lambda_{i,j}\}, \{\mu_{i,j}\}) &= \arg \min_{C_i, \{L_{i,j}\}, \{E_{i,j}\}} \sum_j \|L_{i,j}\|_* + \lambda_{i,j} \|E_{i,j}\|_1 \\ &+ \langle \Lambda_{i,j}, X_{i,j} - C_i - L_{i,j} - E_{i,j} \rangle \\ &+ \frac{\mu_{i,j}}{2} \|X_{i,j} - C_i - L_{i,j} - E_{i,j}\|_F^2 \end{aligned} \quad (2.23)$$

onde  $\mu_{i,j}$  é um escalar,  $\Lambda_{i,j}$  são os multiplicadores de Lagrange e  $\|\cdot\|_F$  é a norma Frobenius<sup>1</sup>.

Resolver diretamente a (2.23) é difícil já que as variáveis dependem umas das outras. Zhang e Li [58] propuseram resolver este problema utilizando o método de descida de coordenadas em blocos, i.e. otimizar um conjunto de parâmetros enquanto os outros ficam fixos. Os parâmetros são divididos em três conjuntos:  $C_i$ ,  $\{L_{i,j}\}$ , e  $\{E_{i,j}\}$ , e o algoritmo é descrito após apresentação de dois lemas [58].

**Lema 2.1.** *A solução para a equação  $C_i = \arg \min_{C_i} \mathcal{L}(C_i, \{\Lambda_{i,j}\}, \{\mu_{i,j}\})$  onde  $\mathcal{L}(C_i, \{\Lambda_{i,j}\}, \{\mu_{i,j}\}) = \sum_j \langle \Lambda_{i,j}, X_{i,j}^c - C_i \rangle + \frac{\mu_{i,j}}{2} \|X_{i,j}^c - C_i\|_F^2$  e  $X_{i,j}^c = X_{i,j} - L_{i,j} - E_{i,j}$ , é:*

$$C_i = \frac{\sum_j \Lambda_{i,j} + \mu_{i,j} X_{i,j}^c}{\sum_j \mu_{i,j}}. \quad (2.24)$$

**Lema 2.2.** *O problema  $\{\{L_{i,j}\}, \{E_{i,j}\}\} = \arg \min_{\{L_{i,j}\}, \{E_{i,j}\}} \mathcal{L}(\{L_{i,j}\}, \{E_{i,j}\}, \{\Lambda_{i,j}\}, \{\mu_{i,j}\})$  pode ser resolvida pela solução de  $\{L_{i,j}, E_{i,j}\}$  para cada  $j \in [1, 2, \dots, N]$  individualmente, onde:*

$$\begin{aligned} \mathcal{L}(\{L_{i,j}\}, \{E_{i,j}\}, \{\Lambda_{i,j}\}, \{\mu_{i,j}\}) &= \sum_j \|L_{i,j}\|_* + \lambda_{i,j} \|E_{i,j}\|_1 \\ &+ \langle \Lambda_{i,j}, X_{i,j}^C - L_{i,j} - E_{i,j} \rangle \\ &+ \frac{\mu_{i,j}}{2} \|X_{i,j}^C - L_{i,j} - E_{i,j}\|_F^2 \end{aligned} \quad (2.25)$$

e  $X_{i,j}^C = X_{i,j} - C_i$ .

---

<sup>1</sup>Também conhecida como norma euclidiana para matrizes. É definida como a raiz quadrada da soma de todos os elementos elevados ao quadrado.

A solução para a matriz comum apresentada no Lema 2.1 é obtida fazendo

$$\frac{\partial \mathcal{L}(C_i, \{\Lambda_{i,j}\}, \{\mu_{i,j}\})}{\partial C_i} = 0, \quad (2.26)$$

motivada pela condição de otimalidade [4, 5, 41]. O Lema 2.2 é baseado na observação que ao minimizar funções convexas (que compõem um somatório) individualmente, minimiza-se também o somatório dessas funções.

O algoritmo proposto por Zhang e Li [58] é mostrado no Algoritmo 2. Uma pequena modificação foi realizada na linha oito do algoritmo. Zhang e Li [58] utilizam  $\frac{\lambda_{i,j}}{\mu_{i,j}}$  como *threshold*. Entretanto, nos testes realizados, este valor não apresentou soluções satisfatórias. De fato, ao observar o algoritmo para solução de PCA Robusto [9, 35, 57], utiliza-se para a determinação da matriz de posto pequeno um *threshold* conforme mostrado em Algoritmo 2. Vale ressaltar que a função objetivo da (2.22) é similar ao PCA Robusto, onde uma matriz é decomposta em duas partes,  $X = L + E$ , com  $L$  sendo de posto pequeno e  $E$  uma matriz esparsa. A diferença principal entre entre PCA Robusto e JSM-MC está na introdução da matriz comum,  $C$ . Isso permite modelar explicitamente a correlação entre todas as imagens do conjunto.

Os parâmetros do Algoritmo 2 utilizado nos testes desta dissertação foram  $\rho = 1, 5$ ,  $\mu_{i,j}^0 = \frac{1,25}{\|X_j\|_2}$  e  $\lambda_{i,j} = \frac{0,5}{\sqrt{w}} \forall j$ , onde  $w$  é a altura da imagem de face, conforme sugerido em [58].

**Algoritmo 2:** JSM-MC [58]

---

```

1: Entrada:  $\rho$ ,  $\{X_{i,j}\}$ ,  $\{\lambda_{i,j}\}$ , e  $\{\mu_{i,j}^0\}$ 
2: Inicialização:  $k = 0$ ,  $C_i^0 = \mathbf{0}$ ,  $\{L_{i,j}^0\} = \{E_{i,j}^0\} = \mathbf{0}$ ,  $\{\Lambda_{i,j}^0\} = \{\frac{X_{i,j}}{\|X_{i,j}\|_F}\}$ 
3: enquanto não convergir faça
4:   para  $j \in [1, 2, \dots, N]$  faça
5:     // Etapa 1: Resolver  $\{E_{i,j}\}$  enquanto  $C_i$  e  $\{L_{i,j}\}$  fixos
6:      $X_{i,j}^C = X_{i,j} - C_i^k$ ;
7:      $E_{i,j}^{k+1} = \mathcal{S}_{\frac{\lambda_{i,j}}{\mu_{i,j}^k}}(X_{i,j}^C - L_{i,j}^k + \frac{1}{\mu_{i,j}^k} \Lambda_{i,j}^k)$ ;
8:     // Etapa 2: Resolver  $\{L_{i,j}\}$  enquanto  $C_i$  e  $\{E_{i,j}\}$  fixos
9:      $(U, \Sigma, V) = \text{svd}(X_{i,j}^C - E_{i,j}^{k+1} + \frac{1}{\mu_{i,j}^k} \Lambda_{i,j}^k)$ ;
10:     $L_{i,j}^{k+1} = U \mathcal{S}_{\frac{1}{\mu_{i,j}^k}}(\Sigma) V^T$ ;
11:     $X_{i,j}^c = X_{i,j} - E_{i,j}^{k+1} - L_{i,j}^{k+1}$ ;
12:   fim
13:   // Etapa 3: Resolver  $C_i$  enquanto  $\{L_{i,j}\}$  e  $\{E_{i,j}\}$  fixos
14:    $C_i^{k+1} = \frac{\sum_j \Lambda_{i,j}^k + \mu_{i,j}^k X_{i,j}^c}{\sum_j \mu_{i,j}^k}$ ;
15:   // Etapa 4: Atualiza  $\Lambda_{i,j}$  e  $\mu_{i,j}$ 
16:   para  $j \in [1, 2, \dots, N]$  faça
17:      $\Lambda_{i,j}^{k+1} = \Lambda_{i,j}^k + \mu_{i,j}^k (X_{i,j} - C_i^{k+1} - L_{i,j}^{k+1} - E_{i,j}^{k+1})$ ;
18:      $\mu_{i,j}^{k+1} = \mu_{i,j}^k \rho$ ;
19:   fim
20:    $k = k + 1$ ;
21: fim
22: Saída:  $C_i$ ,  $\{E_{i,j}\}$  e  $\{L_{i,j}\}$ 

```

---



# Capítulo 3

## O método proposto

Aplicações utilizando o método *Sparse Representation-based Classification* (SRC), apresentado por Yang et al. [56] tornaram-se comum na literatura [17, 44, 45, 55, 60]. No trabalho de Yang et al. [56], o método é utilizado para classificação de faces, e os resultados apresentados são tidos, na literatura, como o estado da arte e por isto o comparamos com o método proposto neste trabalho. O autores Yang et al. [56], propõem um método dito robusto a oclusão, além de testarem o método em imagens sob diferentes condições de iluminação.

A busca por modelos que representem, ou decomponham, um conjunto de sinais são apresentados em [2, 10, 58]. Como consequência, melhores taxas de compressão e extração de características mais relevantes podem ser obtidas. Na abordagem apresentada por Zhang e Li [58], descrita na Subseção 2.4.3, o método JSM-MC foi apresentado, com aplicação em faces. Os resultados apresentados em seus trabalhos indicam que no modelo JSM-MC, a imagem comum obtida sofre pouca influência de oclusões e variações de iluminação presentes no conjunto de faces. Em sua conclusão, os autores atentam para uma potencial aplicação em reconhecimento de faces. Entretanto, nenhuma metodologia foi proposta em seu trabalho para esta aplicação.

O método proposto neste capítulo visa utilizar o JSM-MC para compor o conjunto de treino. Posteriormente, tal conjunto será utilizado no classificador baseado em representação esparsa.

### 3.1 Composição da base de treino

A Figura 3.1 resume os passos para a composição da base de treino do método proposto.

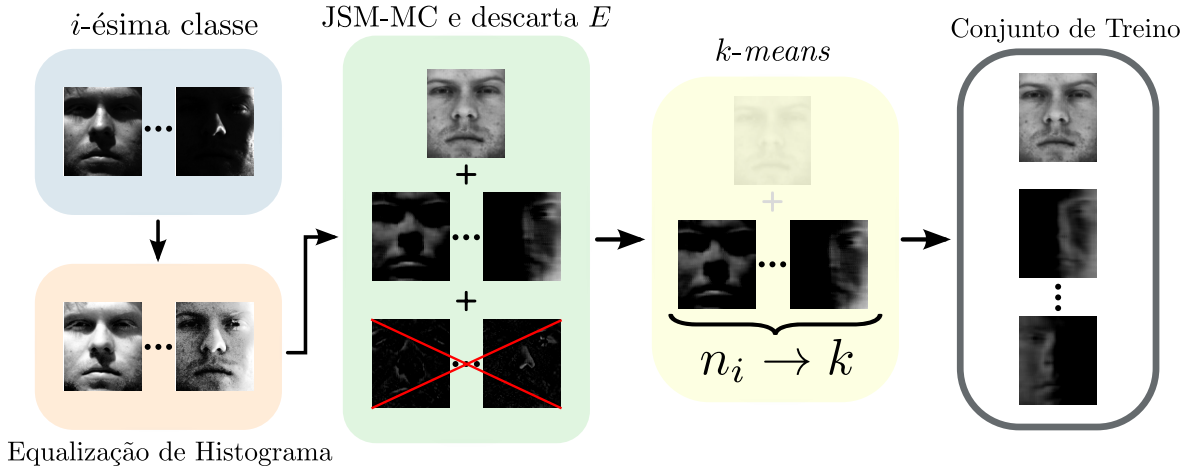


Figura 3.1: Esquema da composição da base de treino

A utilização de um pré-processamento, com objetivo de minimizar os efeitos de iluminação, em sistemas de reconhecimento de face, é uma prática que produz bons resultados na etapa de classificação [19, 30, 36, 48]. Diante de tal observação, o método aqui proposto utiliza da técnica de equalização de histograma como etapa de pré-processamento, afim de minimizar os efeitos de iluminação nas imagens de face objetivando uma maior taxa de acerto na etapa de classificação.

Após escolhido o conjunto de treino (de maneira aleatória, aproximadamente metade das imagens disponíveis na base de dados), e realizado a equalização de histograma em cada imagem, o método JSM-MC é aplicado em cada classe separadamente. Para cada classe, uma imagem comum é obtida, além das matrizes de posto pequeno e esparsa para cada face desta classe. O conjunto de treino é formado apenas pelas imagens comuns e as matrizes de posto pequeno, todas vetorizadas, ignorando-se as matrizes esparsas.

Em testes realizados, a não utilização das matrizes esparsas apresentou uma maior taxa de classificação quando comparada aos testes onde as matrizes esparsas eram utilizadas. Isto ocorre porque componentes esparsas carregam informações (esparsas) referentes à imagem de treino, mas não informações peculiares que definem tal classe. Em outras palavras, as informações esparsas, em geral, podem contribuir em qualquer classe e não somente, como deseja-se, nas classes às quais pertencem.

No intuito de diminuir o tempo na etapa de representação esparsa, uma vez que é necessário resolver o problema de otimização (2.9), realiza-se um agrupamento (*clustering*) das matrizes de posto pequeno vetorizadas em cada classe. Embora seja mais um parâmetro a ser determinado no sistema proposto, nos testes realizados, a utilização de

um número equivalente à metade das faces de treino, por classe, disponíveis apresentou resultados satisfatórios na classificação de faces sem oclusão. Portanto, nos testes e resultados que envolvem classificação de faces sem oclusão, 15 grupos são formados por classe.

O Algoritmo 3 resume a etapa de elaboração da base de treino.

---

**Algoritmo 3:** Composição da base de treino

---

- 1: **Entrada:** Imagens de treino  $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_k\}$  para  $k$  classes, onde  $\mathcal{X}_i = \{X_{i,1}, \dots, X_{i,n_i}\}$ ,  $i = 1, \dots, k$  e  $n_i$  é o número de imagens de treino da  $i$ -ésima classe.
  - 2: Equalização de histograma para cada  $X_{i,j}$ , com  $i = 1, \dots, k$  e  $j = 1, \dots, n_i$ .
  - 3: Aplicar o método JSM-MC (Algoritmo 2) em cada  $\mathcal{X}_i$ . Descartar as matrizes esparsas e montar a matriz de treino  $A = [A_1 \ \dots \ A_k]$  onde  $A_i = [c_i \ l_{i,1} \ \dots \ l_{i,n_i}]$ .  $c_i$  e  $l_{i,j}$  são as matrizes  $C_i$  e  $L_{i,j}$  vetorizadas, respectivamente.
  - 4: Opcionalmente, utilizar o agrupamento (*clustering*) para cada  $A_i^- = [l_{i,1} \ \dots \ l_{i,n_i}]$  (sem a componente comum,  $c_i$ , de  $A_i$ ).
  - 5: **Saída:** Matriz  $A$  (com as colunas normalizadas para ter norma  $\ell_2$  unitária).
- 

## 3.2 O algoritmo de classificação

A Figura 3.2 resume os passos para classificação de uma imagem de teste.

As etapas descritas na seção anterior determinam a base de treino que será utilizada no processo de classificação. De posse de tal conjunto, para uma nova face de teste, aplica-se a equalização de histograma nesta imagem. Então, a imagem resultante deste pré-processamento é vetorizada e obtém-se sua representação esparsa resolvendo o problema de otimização (2.9). Para determinar a classe da face de teste, Yang et al. [56] propõe a utilização dos resíduos como critério de classificação, conforme (2.10). Ao invés disso, no critério de classificação aqui proposto, a classe é determinada pela maior norma- $\ell_1$  das funções características, i.e.,

$$\arg \max_i u_i \doteq \| \delta_i(\hat{\mathbf{x}}_1) \|_1 . \quad (3.1)$$

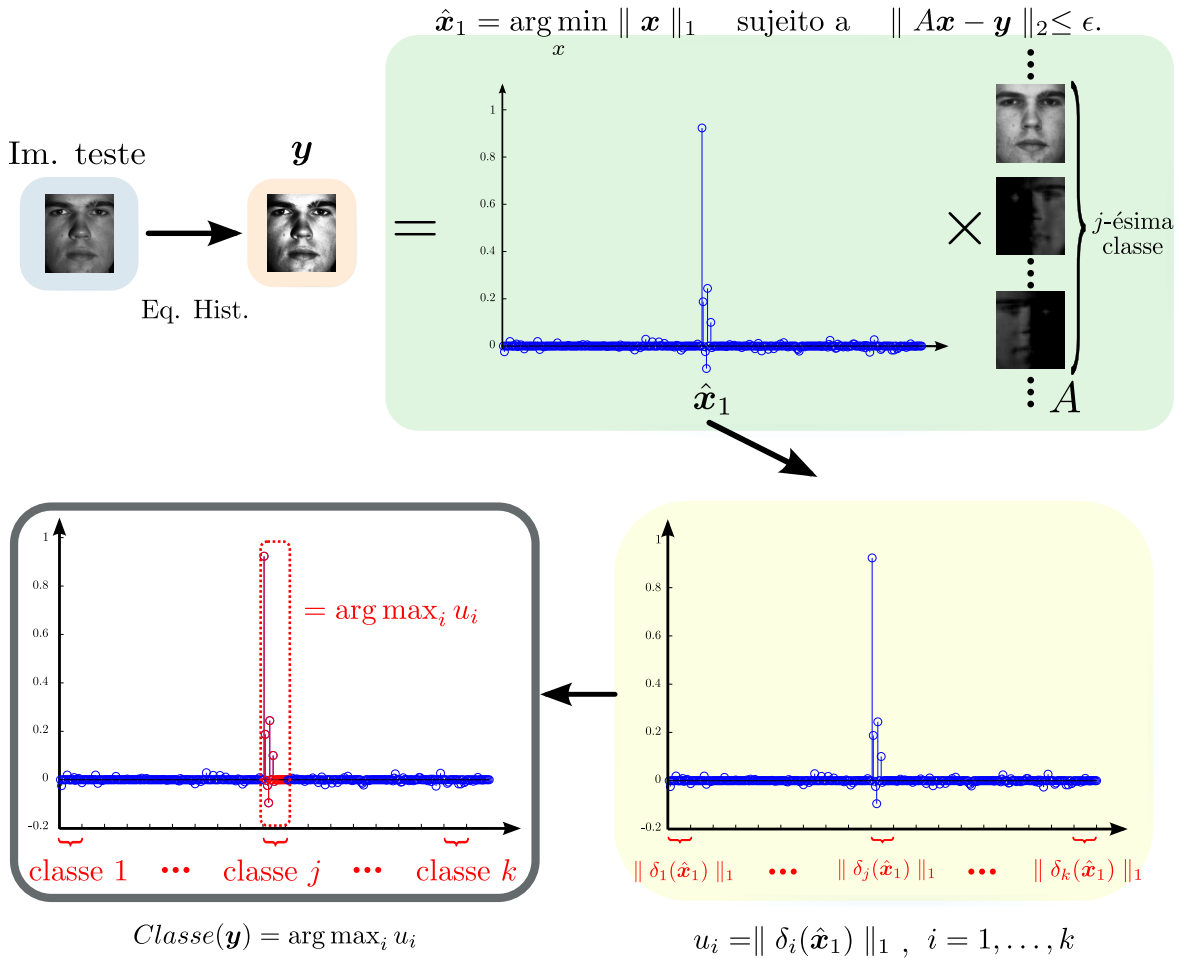


Figura 3.2: Esquema de classificação

Em outras palavras, procura-se a classe cujos respectivos coeficientes da representação esparsa mais contribuem para representação da imagem de teste.

Conforme observado em [56], idealmente, os elementos diferentes de zero de  $\hat{\mathbf{x}}_1$  estarão associados às colunas de  $A$  de uma única classe  $i$ , o que não acontece no sistema real devido a ruídos e erros de modelagem, ocasionando alguns elementos diferentes de zero associados à múltiplas classes. Entretanto, espera-se que tais elementos sejam pequenos, e portanto não contribuem fortemente para suas respectivas classes ao utilizar a classificação proposta (3.1).

Para motivar a utilização do critério (3.1) e não a (2.10), considera-se as três imagens mostradas na Figura 3.3 e faz-se os seguintes questionamentos:

- Entre a Figura 3.3b e Figura 3.3c, qual se assemelha (menor resíduo) mais à Figura 3.3a?

- Entre a Figura 3.3b e Figura 3.3c, qual imagem pode representar a Figura 3.3a apenas pela variação no *offset* (adição ou subtração de uma constante aos níveis de cinza de todos os *pixels* da imagem) e na escala de nível de cinza?

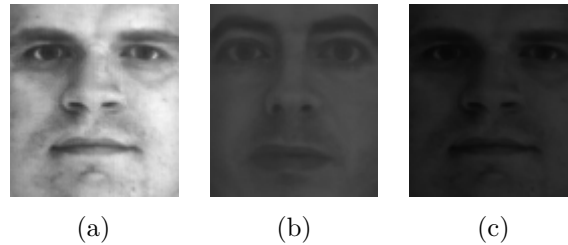


Figura 3.3: Exemplos de três imagens de face.

Para auxiliar a resposta dessas perguntas, as imagens são vetorizadas e exibidas na Figura 3.4.

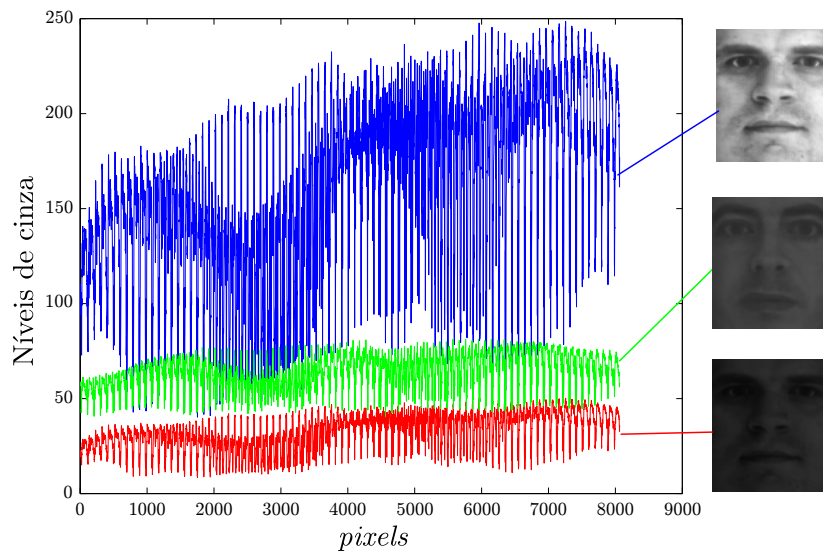


Figura 3.4: Gráfico das três imagens vetorizadas.

As Figura 3.3a e Figura 3.3c são imagens da mesma pessoa. Entretanto, do ponto de vista de menor resíduo, a imagem que mais se assemelha à Figura 3.3a é a Figura 3.3b. Esta observação responde à primeira pergunta feita. Para responder a segunda pergunta, observa-se em gráficos separados e em diferentes escalas da Figura 3.3a e Figura 3.3c vetorizadas conforme mostra a Figura 3.5. Logo, a menos de uma mudança de escala e variação de *offset*, a Figura 3.5a é muito similar a Figura 3.5b e portanto, a resposta para segunda pergunta é: a Figura 3.3c melhor representa a Figura 3.3a. Observe ainda que a utilização de diferentes escalas entre as imagens no exemplo pode ser justificada pela utilização da função característica nos critérios de classificação. Assim,

como no critério onde os resíduos são utilizados compara-se a imagem de teste com a reconstrução utilizando a função característica, naturalmente, as escalas das imagens serão diferentes. Por esse exemplo é possível observar que a utilização dos resíduos como critério para determinação da classe da imagem de teste pode não revelar a real classe de uma imagem ao passo que uma medida da representatividade, por classe, de uma imagem de teste revelaria a real classe. Assim, opta-se pela utilização de (3.1) como critério de classificação. No Capítulo 5, isso fica ainda mais evidente, já que nos resultados apresentados, a utilização de (3.1) apresenta melhor desempenho quando comparada à (2.10).

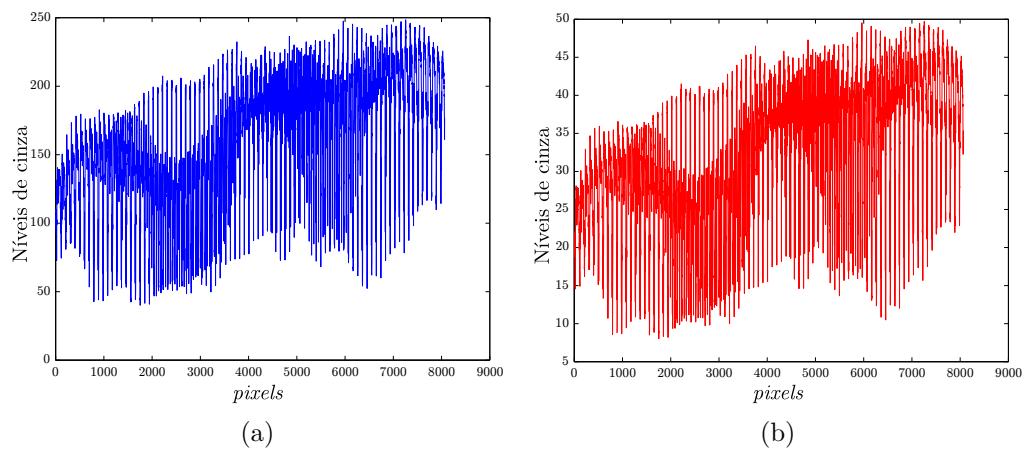


Figura 3.5: (a) Gráfico da Figura 3.3a vetorizada e (b) gráfico da Figura 3.3c vetorizada.

Vale ressaltar que o método proposto para classificação corrobora com a não utilização das matrizes esparsas discutido na seção anterior. Por serem esparsas, suas contribuições poderiam ser mais significativas em múltiplas classes, tornando o vetor de representação  $\hat{x}_1$  menos esperso.

O Algoritmo 4 resume o procedimento de classificação.

### 3.2.1 Versão estendida

Para imagens com oclusão, a versão estendida apresenta melhor resultados. Isto se deve ao fato do erro ser considerado no modelo do problema de otimização. Para isso, substitui-se a etapa 3 do Algoritmo 4 pelo problema de otimização estendido (2.15). Além disso, para possibilitar a execução do programa na memória do MATLAB<sup>®</sup>, reduz-se as imagens de  $192 \times 168$  para  $96 \times 84$  *pixels*.

O Algoritmo 5 resume o procedimento de classificação.

---

**Algoritmo 4:** Representação esparsa e classificação
 

---

- 1: **Entrada:** Matriz das amostras de treino  $A$ . Amostra de teste  $\mathbf{y}$ . Tolerância de erro  $\epsilon > 0$ .
- 2: Aplicar equalização de histograma na imagem de teste  $\mathbf{y}$  e normalizar  $\mathbf{y}$  para ter norma  $\ell_2$  unitária.
- 3: Resolver o problema de minimização- $\ell_1$  (2.12):

$$\hat{\mathbf{x}}_1 = \arg \min_x \|\mathbf{x}\|_1 \quad \text{sujeito a} \quad \|R\mathbf{A}\mathbf{x} - R\mathbf{y}\|_2 \leq \epsilon.$$

- 4: Determinar  $u_i = \|\delta_i(\hat{\mathbf{x}}_1)\|_1$  para  $i = 1, \dots, k$ .
  - 5: **Saída:**  $Classe(\mathbf{y}) = \arg \max_i u_i$ .
- 

---

**Algoritmo 5:** Representação esparsa e classificação - versão estendida
 

---

- 1: **Entrada:** Matriz das amostras de treino  $A$  onde as imagens são de  $96 \times 84$  *pixels*. Amostra de teste  $\mathbf{y}$  reduzida para  $96 \times 84$  *pixels*.
- 2: Aplicar equalização de histograma na imagem de teste  $\mathbf{y}$  e normalizar  $\mathbf{y}$  para ter norma  $\ell_2$  unitária.
- 3: Resolver o problema de minimização- $\ell_1^e$  (2.15):

$$\hat{\mathbf{w}}_1 = \arg \min \|\mathbf{w}\|_1 \quad \text{sujeito a} \quad \mathbf{y} = B\mathbf{w}.$$

onde  $\hat{\mathbf{w}}_1 = [\hat{\mathbf{x}}_1 \quad \hat{\mathbf{e}}_1]^T$ ,  $\mathbf{w} = [\mathbf{x} \quad \mathbf{e}]^T$  e  $B = [A \quad I]$ .

- 4: Determinar  $u_i = \|\delta_i(\hat{\mathbf{x}}_1)\|_1$  para  $i = 1, \dots, k$ .
  - 5: **Saída:**  $Classe(\mathbf{y}) = \arg \max_i u_i$ .
-

---

Para todos os teste presentes nesta dissertação que envolvem a solução do Algoritmo 4 o valor de  $\epsilon$  utilizado foi de 0,05 conforme sugerido em [54].



# Capítulo 4

## Testes

Neste capítulo apresenta-se os testes de cada etapa do algoritmo. Uma breve comparação entre a equalização de histograma e filtragem homomórfica é feita, justificando-se a utilização da equalização de histograma neste trabalho. Ainda neste capítulo, apresenta-se os testes para a decomposição usando JSM-MC, além da motivação para a não utilização das matrizes esparsas para a composição da base de treino. Testes com o agrupamento também são apresentados bem como seus impactos nas matrizes de posto pequeno. Por fim, realiza-se testes com diferentes oclusões para melhor entender seus efeitos nas imagens de face.

### 4.1 Equalização de histograma

Como dito anteriormente, esta etapa de pré-processamento é feita na esperança de minimizar os efeitos de iluminação nas imagens de face. Vários métodos podem ser aplicados nessa etapa (e.g. [19, 30, 36, 48]) para obter uma melhora na taxa de reconhecimento de padrões. Ao realizar um teste utilizando filtragem homomórfica, descartou-se o uso desta, uma vez que o resultado da filtragem depende fortemente dos parâmetros do filtro, e para diferentes condições de iluminação nas imagens, diferentes parâmetros são necessários para obter uma melhora de contraste nas imagens. Isso motivou a busca por um método de pré-processamento que não dependesse de um parâmetro especial para diferentes condições de iluminação nas imagens de face. Nesse sentido, a equalização de histograma apresentou um resultado satisfatório sem a necessidade de sintonizar parâmetro algum. Além disso, trabalhos realizados por Mendonça et al. [43] e Mendonça [42] indicam que para os métodos de classificação

utilizados em seus trabalhos (auto-faces, PCA com rede neural e *wavelets* com rede neural), a filtragem homomórfica pouco melhorava a taxa de reconhecimento de faces.

Como pode ser observado na Figura 4.1, o contraste nas imagens processadas apresenta uma melhora significativa ao utilizar a equalização de histograma. Verifica-se também que nos casos de iluminações muito irregulares, o método apresenta deterioração da imagem. Contudo, na etapa de tratamento no domínio esparso, é possível observar que este problema é minimizado, uma vez que tal deterioração apresenta uma característica esparsa. Após a decomposição usando o JSM-MC estas componentes são expressas pelas matrizes esparsas e, como descrito anteriormente, não são utilizadas para compor a matriz de treino.



Figura 4.1: Resultado da equalização de histograma para quatro imagens de face. As imagens superiores são as faces originais e as inferiores são os resultados após o pré-processamento

## 4.2 Decomposição usando JSM-MC

Utilizando 32 imagens (incluindo as imagens utilizadas no teste de equalização de histograma), realizou-se a decomposição utilizando o método JSM-MC. O resultado da imagem comum pode ser observado na Figura 4.2. Tal imagem, aparentemente, não sofre com as diferentes iluminações do conjunto de faces utilizado.



Figura 4.2: Imagem comum obtida para o conjunto de faces utilizadas.

Embora a equalização de histograma melhore o contraste da imagem, em severas iluminações observa-se um efeito ruidoso indesejado nas imagens. A Figura 4.3 mostra as demais componentes resultantes do método JSM-MC. É possível observar que parte do efeito indesejado, que possui característica esparsa, causado pela equalização de histograma é expresso nas imagens referentes às matrizes esparsas.

Para melhor verificar este efeito, reconstrói-se as imagens utilizando apenas a componente comum e as matrizes de posto pequeno conforme mostra Figura 4.4. As imagens reconstruídas, sem as matrizes esparsas, de fato apresentam uma menor degradação recorrente da equalização de histograma. Essa observação ratifica a não utilização das matrizes esparsas na composição da base de treino.

### 4.3 Agrupamento (*clustering*) para redução da base de treino

Esta etapa é feita para reduzir o conjunto de treino e a complexidade no momento da representação esparsa. Para isto, o método *k-means* é utilizado. Utiliza-se um método denominado Sammon *map* [47], concebido para visualização de dados de alta dimensão, para verificar os agrupamentos formados pelo *k-means*[20]. Sammon *map* mapeia dados de alta dimensão em duas ou três dimensões de maneira a preservar as distâncias entre as amostras no espaço mapeado.

A Figura 4.5 mostra o resultado do mapeamento usando o Sammon *map* para uma classe com 32 faces e aplicação do *k-means* para 15 grupos. Observa-se que o agrupamento formado é coerente, contudo, alguns grupos se “misturam” (e.g. grupo um e três). Isto se deve ao fato da drástica redução de dimensão imposta para sua visualização e/ou os grupos formados realmente serem próximos. Apesar de não ser feito neste trabalho, pois deseja-se deixar o sistema o mais automático possível, o



Figura 4.3: Imagens correspondentes às matrizes de posto pequeno e esparsas oriundas da decomposição usando o método JSM-MC. As imagens superiores são algumas das imagens usadas (após a equalização de histograma). As imagens localizadas no meio são referentes as matrizes de posto pequeno e as imagens inferiores correspondem às matrizes esparsas.

Sammon *map* pode ser utilizado para determinar a quantidade de agrupamentos que devem ser gerados pelo *k-means*, ou mesmo a utilização de outros métodos para realizar o agrupamento como o *isodata*.

Cada agrupamento é substituído pelo centróide correspondente. Nos testes e resultados com imagens de teste sem oclusão apresentados neste trabalho consideram-se 15 *clusters*. A Figura 4.6 mostra o centróide e as imagens correspondentes ao *cluster* um.

Na Figura 4.7 é possível observar que de fato os grupos um e três são semelhantes, conforme indicado no Sammon plot. Ainda no Sammon plot, é possível obter uma ideia da iluminação nas imagens. Os grupos localizado na parte esquerda do gráfico



Figura 4.4: Comparação entre as imagens originais (superior), com histograma equalizado (meio) e reconstruídas sem a influência das matrizes esparsas (inferior).

da Figura 4.5 possuem níveis de cinza mais baixos na parte esquerda da imagem (Figura 4.6). Os grupos localizados na parte direita do gráfico possuem níveis de cinza mais baixos na parte direita da imagem (Figura 4.8), e os grupos localizados no centro apresentam níveis de cinza mais baixos no centro da imagem (Figura 4.9).

Essa observação motivou a redução do número de agrupamentos na classificação de faces com oclusões na base de teste, uma vez que a versão estendida do problema de otimização é utilizada e seu custo computacional é maior. Portanto, 4 grupos são utilizados, ao invés de 15, nos resultados que envolvem oclusão na base de teste e utilizam o problema de otimização estendida.

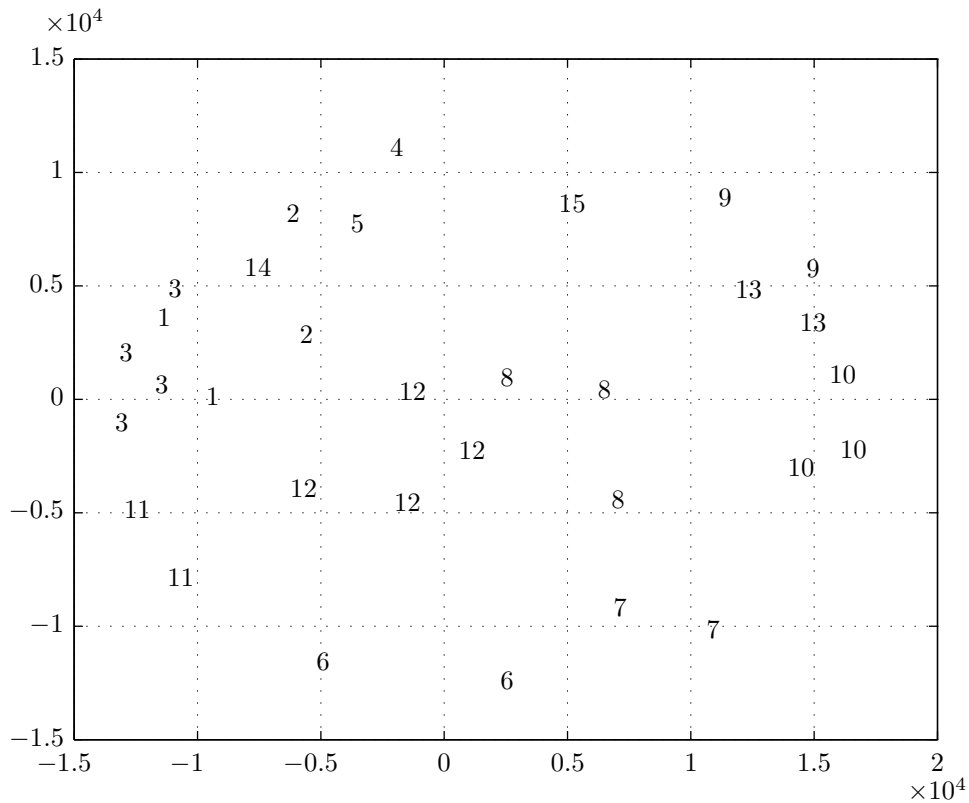


Figura 4.5: Sammon plot para visualização do agrupamento realizado utilizando o *k-means*. Cada agrupamento (*cluster*) é substituído pelo centróide.



Figura 4.6: *Cluster 1*. Na imagem da esquerda está o centróide e as outras imagens correspondem ao *cluster 1*.

## 4.4 Tipos de oclusões

Para verificar os efeitos dos diferentes tipos de oclusões em imagens de face testadas neste trabalho, o seguinte teste foi realizado: na imagem de face, indicada na Figura 4.10, insere-se as oclusões. O classificador, em sua versão estendida, é utilizado para verificar o efeito das oclusões na classificação. Para os diferentes tipos de oclusões, as estimativas dos vetores de representação esparsa, a estimação do erro, a



Figura 4.7: *Cluster 3*. Na imagem da esquerda está o centróide e as três imagens correspondem ao *cluster 3*.



Figura 4.8: *Cluster 10*. Na imagem da esquerda está o centróide e as demais imagens correspondem ao *cluster 10*.



Figura 4.9: *Cluster 8*. Na imagem da esquerda está o centróide e as demais imagens correspondem ao *cluster 8*.

imagem reconstruída utilizando a solução encontrada e a reconstrução utilizando apenas os elementos da classe correta da solução de otimização encontrada são exibidas.

#### 4.4.1 Oclusão aleatória

Neste trabalho, chama-se oclusão aleatória como a substituição dos valores de nível de cinza de *pixels* (escolhidos de forma aleatória) da imagem de face por amostras



Figura 4.10: Imagem de teste utilizada para verificar o efeito da oclusão em blocos na estimação do erro e do vetor de pesos

independentes e identicamente distribuídas de uma distribuição uniforme no intervalo de  $[0, 255]$ . Foram realizadas oclusões aleatórias nas taxas de 10%, 25%, 50% e 75%. Os resultados para o teste podem ser observados na Figura 4.11.

É possível observar que os vetores da solução do problema de otimização são de fato esparsos, mesmo para oclusões aleatórias de 75%. Entretanto, para tal taxa de oclusão, a contribuição das outras componentes da base de treino começam a fazer mais efeito na estimação do vetor de representação esparsa. Vale ressaltar que para uma oclusão de 75%, até para o ser humano, a determinação da real identidade da imagem de face se torna uma tarefa difícil.

#### 4.4.2 Oclusão em blocos

Neste teste, verifica-se o efeito de uma oclusão em blocos. Para isso, parte da imagem é substituída por um bloco onde todos seus elementos são iguais a zero. Os resultados para oclusão de 10%, 25%, 50% e 75% de oclusão são mostrados na Figura 4.12.

Ao observar os vetores de representação esparsa, mostrados na Figura 4.12, verifica-se que, de maneira geral, estes são menos esparsos do que os vetores encontrados para oclusão aleatória. Em especial, para oclusões iguais e maiores que 50%, o sistema já não é mais capaz de determinar corretamente a identidade da imagem de face.

Ainda na oclusão em blocos, outro teste é realizado, contudo modificando o conteúdo da oclusão. Parte da imagem (em posição aleatória) é substituída por uma imagem que não é face, neste caso, utiliza-se a imagem de um babuíno na oclusão em blocos. Foram realizadas oclusões nas taxas de 10%, 25%, 50% e 75%. Os resultados para o teste podem ser observados na Figura 4.13.

Para oclusões em blocos com imagens não relacionadas (neste caso o macaco babuíno), observa-se que os vetores de representação esparsa encontrados também são,



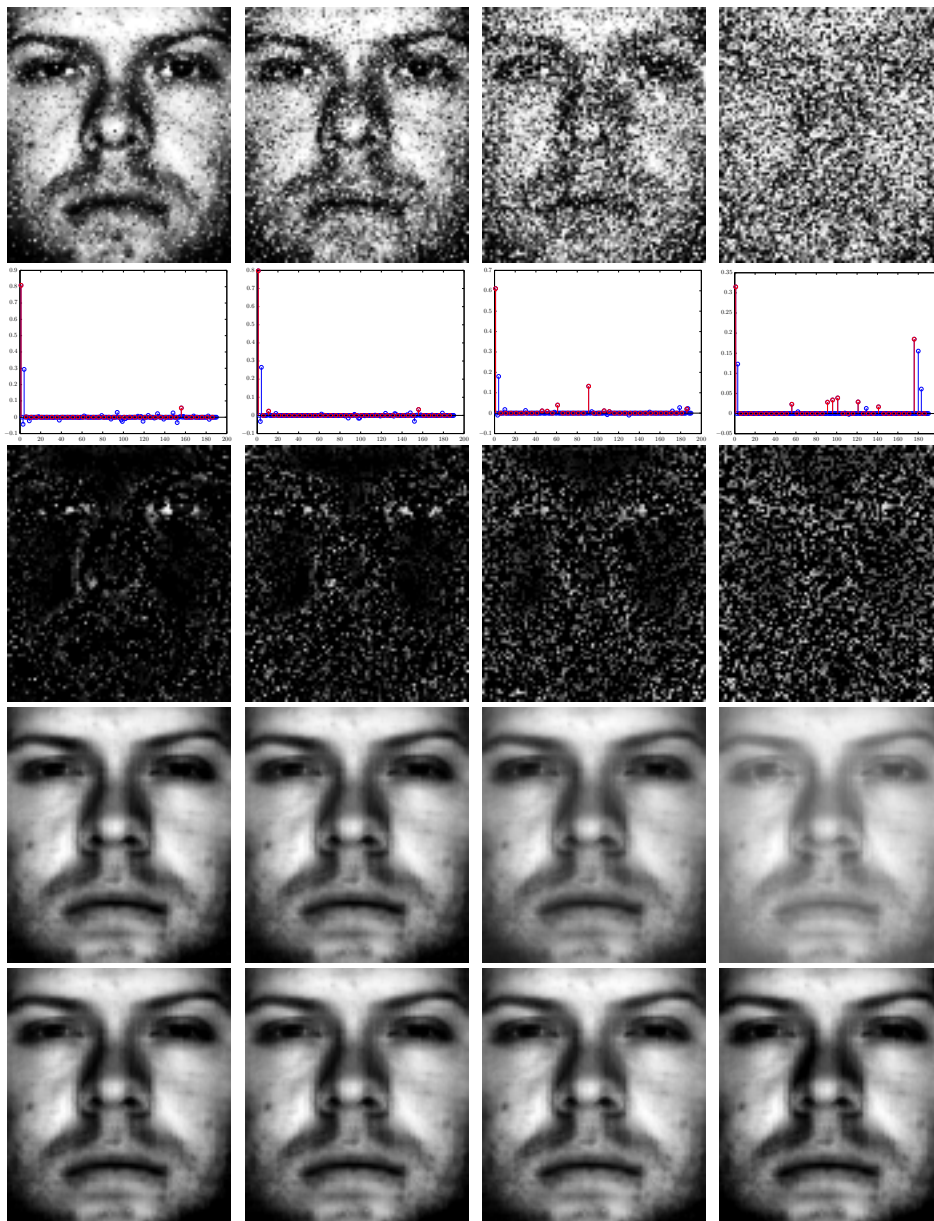


Figura 4.11: Efeito da oclusão aleatória na estimação do erro e do vetor de pesos. A primeira coluna referente a 10% de ruído aleatório e a última 75% de ruído (25% e 50% de ruído para a segunda e terceira coluna respectivamente). A primeira linha são as imagens com o ruído, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste

visualmente, menos esparsos do que as soluções obtidas no caso de oclusão aleatória. Entretanto, tais vetores são mais esparsos quando comparados com o teste de oclusão usando um bloco com seus elementos iguais a zero. Isso motivou uma melhor investi-

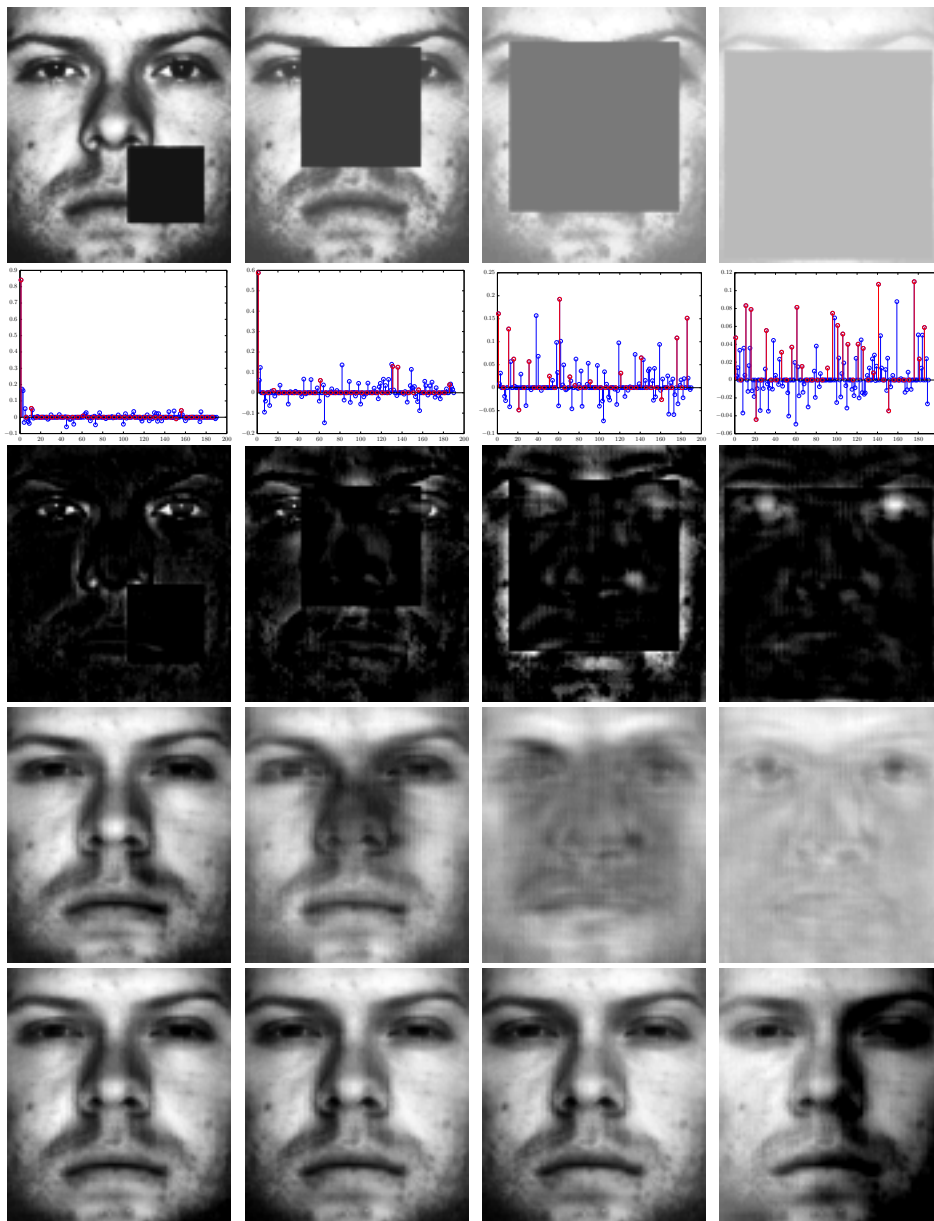


Figura 4.12: Efeito da oclusão em blocos em posição aleatória, usando um bloco com todos os elementos iguais a zero, na estimação do erro e do vetor de pesos. A primeira coluna referente a 10% de ruído aleatório e a última 75% de ruído (25% e 50% de ruído para a segunda e terceira coluna respectivamente). A primeira linha são as imagens com o ruído após a equalização de histograma, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste

gação do efeito da aleatoriedade da oclusão em blocos na estimação do erro e do vetor

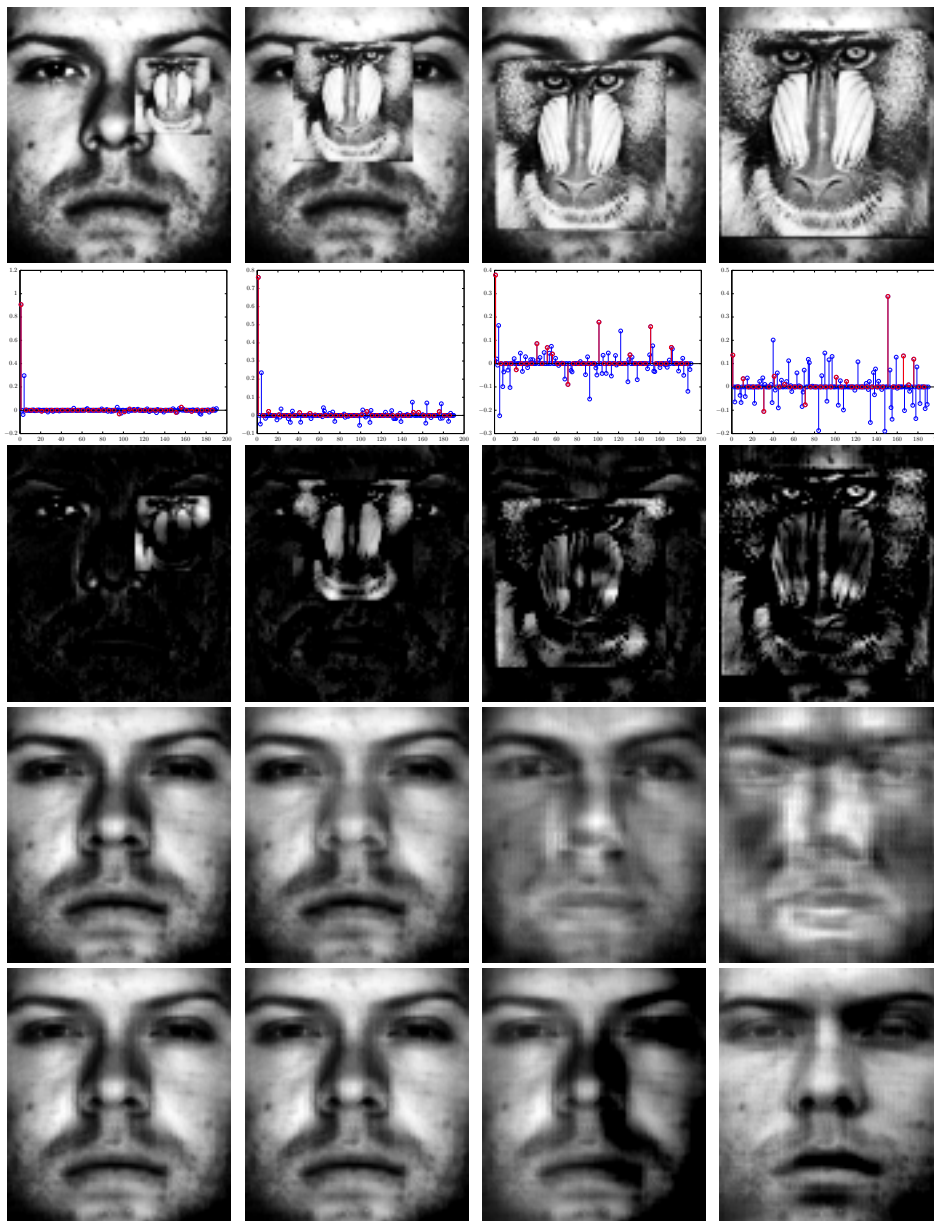


Figura 4.13: Efeito da oclusão em blocos em posição aleatória e usando uma imagem não face na estimação do erro e do vetor de pesos. A primeira coluna referente a 10% de ruído aleatório e a última 75% de ruído (25% e 50% de ruído para a segunda e terceira coluna respectivamente). A primeira linha são as imagens com o ruído, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste

de representação esparsa.

### 4.4.3 Oclusão em blocos com ruído no bloco

Para verificar o efeito da aleatoriedade da oclusão em uma imagem, uma oclusão em bloco, representando 50% da imagem, em uma posição fixa é inserida na imagem. Tal oclusão possui nível de cinza zero, sendo uma imagem preta. Dentro desse bloco, um ruído aleatório (i.i.d de uma distribuição uniforme no intervalo de  $[0, 255]$ ) é inserido em diferentes taxas (25%, 50%, 75% e 100%). Os resultados para o teste podem ser observados na Figura 4.14.

É possível observar que para a oclusão em blocos, sem adição de ruído no bloco, a solução obtida no problema de otimização foi menos esparsa do que nos outros casos (onde havia ruído no bloco). Além disso, ao observar as imagens reconstruídas utilizando tal solução, à medida que o bloco se torna mais aleatório<sup>1</sup>, menor é o efeito do bloco na imagem reconstruída (neste caso, menor efeito não significa uma imagem mais próxima da desejada, significa apenas que a imagem reconstruída não apresenta a característica aleatória do ruído inserido nem do bloco).

Portanto, as seguintes conclusões podem ser obtidas neste teste:

- ao aumentar a aleatoriedade da oclusão (no bloco) um vetor mais esparsa é encontrado;
- quando o bloco de oclusão não possui nenhum tipo de aleatoriedade, a oclusão é de posto pequeno, e portanto, mais fácil é de representá-la pelas matrizes de posto pequeno “ $L$ ” de (2.21). Por isso, menos provável de ser representada no vetor de erro, e menos esparsa será a solução do problema de otimização;
- à medida que o bloco de oclusão se torna mais aleatório, sua representação pelas matrizes “ $C$ ” e “ $L$ ” de (2.21) se torna menos provável, e portanto, tal oclusão será melhor estimada pelo erro, tornando a solução da otimização mais esparsa. Apesar dos teste envolverem apenas ruído aleatório (i.i.d. de uma distribuição uniforme), acredita-se que esta conclusão ainda seja válida para ruídos com outras distribuições (e.g. gaussiana), já que estes dificilmente terão posto pequeno.

### 4.4.4 Oclusão em blocos sem equalização de histograma

O último teste de oclusão refere-se ao mesmo teste anterior, contudo sem a etapa da equalização de histograma. Isto é feito para verificar o efeito da equalização de

<sup>1</sup>Neste caso, mais aleatório significa apenas um maior número de *pixels* substituídos por um ruído aleatório

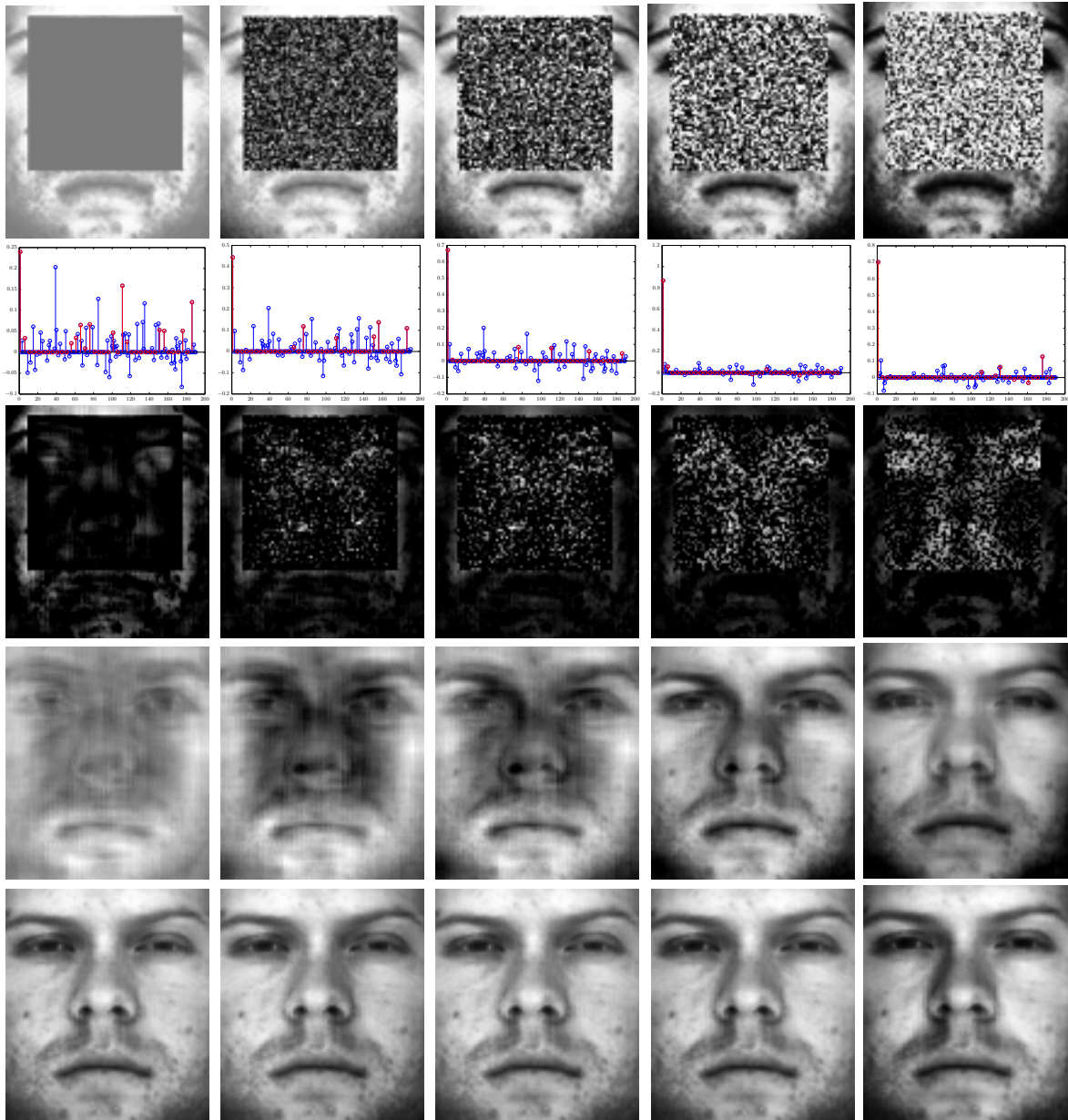


Figura 4.14: Efeito da oclusão em blocos na estimação do erro e do vetor de pesos. A primeira coluna referente a 0% de ruído dentro do bloco e a última 100% de ruído dentro do bloco (incremento de 25% de ruído no bloco por coluna). A primeira linha são as imagens com o ruído, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste.

histograma na estimação do erro e do vetor de representação esparsa quando as imagens de testes possuem oclusão.

Os resultados para esse teste são mostrados na Figura 4.15. Os resultados indicam

---

um comportamento similar ao apresentado no teste com a etapa de equalização de histograma. Isto é, ao aumentar a aleatoriedade no bloco de oclusão, o vetor de representação esparsa se torna mais esparso. Contudo, observa-se que para oclusão em bloco com 100% de aleatoriedade no bloco, apesar do vetor de representação ser esparso, a identidade correta da imagem de face não é obtida. A reconstrução da imagem usando vetor de representação esparsa indica que duas classes prevalecem na reconstrução, já que é possível observar parte do nariz de uma segunda classe. Isso pode ser observado no vetor de representação esparsa também, uma vez que duas componentes comuns possuem os maiores valores absolutos dentro do vetor em intensidade similar.

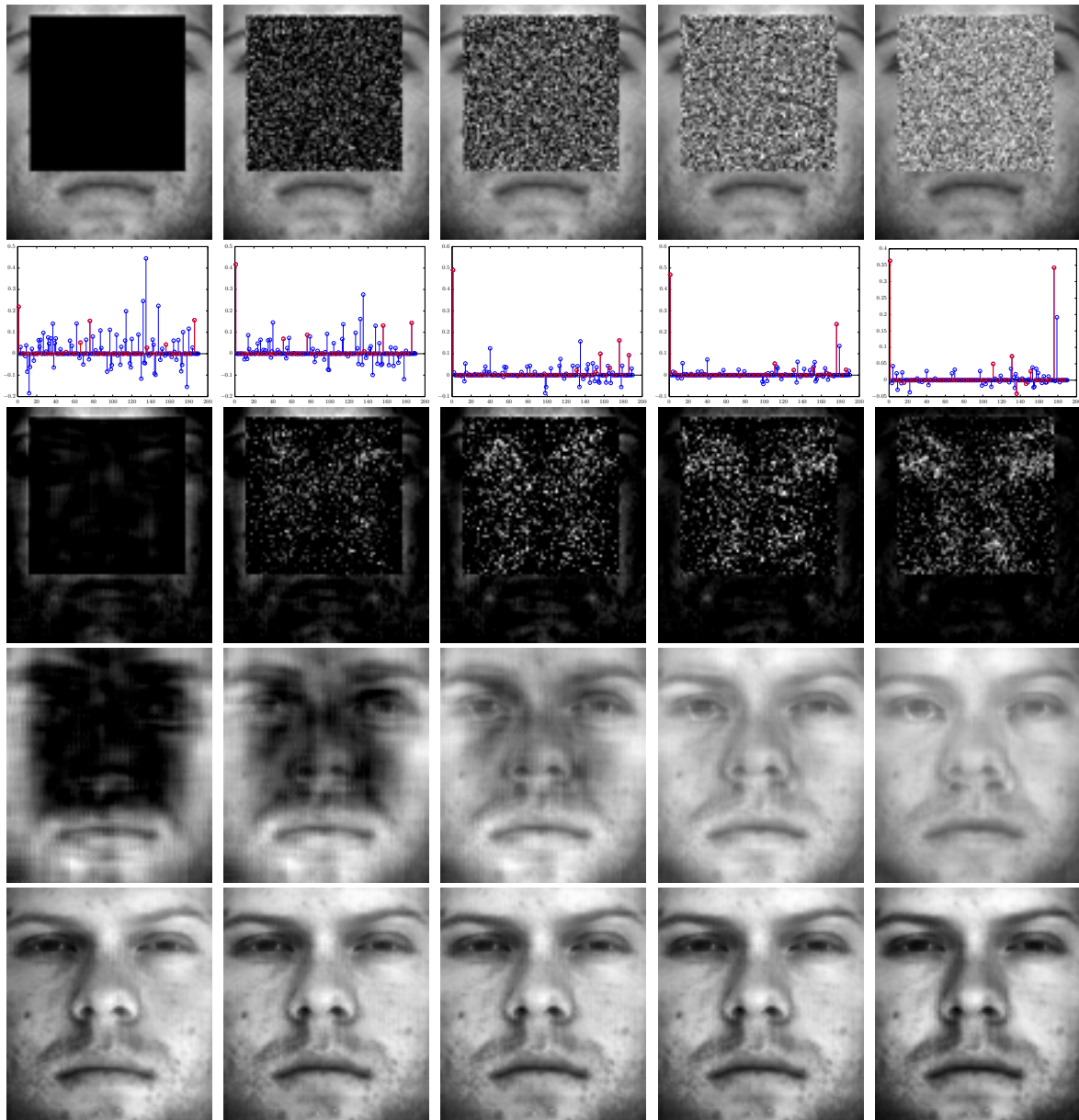


Figura 4.15: Efeito da oclusão em blocos na estimação do erro e do vetor de pesos (sem a etapa de equalização de histograma). A primeira coluna referente a 0% de ruído dentro do bloco e a última 100% de ruído dentro do bloco (incremento de 25% de ruído no bloco por coluna). A primeira linha são as imagens com o ruído, a segunda mostram as soluções para o problema de otimização usado para a classificação, a terceira são os erros estimados, na quarta são as imagens reconstruídas utilizando a solução encontrada e a última representa as imagens reconstruídas utilizando apenas os pesos referentes à classe real da face de teste

# Capítulo 5

## Resultados

Neste capítulo mostra-se os resultados obtidos na classificação de faces. A base de dados utilizada possui diferentes condições de iluminação. Diferentes oclusões artificiais são feitas para verificar o comportamento do sistema quando a imagem de teste ou de treino possui oclusão. Além disso, os resultados para rejeição de imagens de teste que não correspondem à uma classe da base de treino são apresentados.

### 5.1 Imagens de teste sem oclusões

Para os resultados apresentados nesta seção, utilizou-se a seguinte metodologia: escolhe-se, de maneira aleatória, aproximadamente metade do número de faces disponíveis por classe como conjunto de treino e o restante como conjunto de teste (total de 1207 imagens de teste, aproximadamente 32 imagens por classe), este procedimento será chamado de realização. Tal procedimento é repetido dez vezes, totalizando dez realizações. Para cada realização, utiliza-se  $d = 30, 56, 120, 200$  e  $300$  características e uma matriz de transformação  $R$ , conforme definida anteriormente, diferente para cada  $d$ . O motivo pelo qual adota-se tal metodologia é para garantir que os resultados e conclusões não sejam polarizadas por um conjunto de dados especial ou uma matriz  $R$  específica. Os parâmetros utilizados nos algoritmos foram conforme descrito nas páginas 38 e 47.

A Tabela 5.1 mostra as taxas de acerto utilizando o método proposto bem como o tempo médio obtido para classificar uma nova amostra. Uma comparação entre o classificador (Algoritmo 4) usando o critério de classificação baseado na  $\|\delta_i(\hat{\mathbf{x}}_1)\|_1$ , indicado por  $(u)$ , e na  $\|\mathbf{y} - A\delta_i(\hat{\mathbf{x}}_1)\|_2$ ,  $(r)$ , é feita. Além disso, outros dois critérios de classificação foram testados: em  $(u^c)$  uma restrição em  $(u)$  é adicionada, onde



o elemento do vetor de representação esparsas associado à componente comum da classe deve ser diferente de zero. Portanto, antes de calcular a norma  $\ell_1$  da função característica,  $\|\delta_i(\hat{\mathbf{x}}_1)\|_1$ , verifica-se o elemento do vetor de representação esparsa associado à componente comum da  $i$ -ésima classe e caso este seja zero, tal classe é desconsiderada na classificação; em ( $f$ ), a combinação dos três classificadores é feita. Neste caso, uma votação é feita entre os três critérios ( $u$ ), ( $r$ ) e ( $u^c$ ). Logo, a classe que tiver pelo menos dois critérios atendidos é atribuída como classe da imagem de teste. Caso os três critérios retornem classes distintas, um dos critérios é escolhido para atribuir a classe da imagem de teste. Então, na primeira ocorrência de divergência dos três critérios, ( $u$ ) é escolhido como critério para determinar a classe, na segunda ocorrência, ( $r$ ) é escolhido, na terceira ( $u^c$ ) é escolhido e nas próximas repete-se essa ordem de escolha. Isto é feito para garantir que o resultado não seja tendencioso à nenhum critério específico.

Os melhores resultados estão destacados em negrito na Tabela 5.1. Além disso, a menor taxa de acerto obtida de todas as realizações é indicada por ‘min’ na tabela, a maior por ‘max’ e a média por ‘med’, para todos os critérios testados. O tempo indicado na tabela está em segundos e representa o tempo médio para classificação de uma imagem de teste.

Tabela 5.1: Taxas de acerto obtidas usando o método descrito

| dim (d)         | 30             | 56             | 120            | 200            | 300            |
|-----------------|----------------|----------------|----------------|----------------|----------------|
| $(u^c)$ min [%] | 66,0315        | 91,3007        | 98,0116        | 99,2543        | 99,2543        |
| $(u^c)$ med [%] | 71,6487        | 92,6761        | <b>98,7821</b> | 99,5526        | 99,5775        |
| $(u^c)$ max [%] | 76,9677        | 94,6976        | 99,2543        | 99,8343        | 99,7514        |
| $(u)$ min [%]   | 65,7829        | 91,4664        | 98,0116        | 99,2543        | 99,2543        |
| $(u)$ med [%]   | <b>71,8641</b> | <b>92,7672</b> | 98,749         | 99,5443        | <b>99,5857</b> |
| $(u)$ max [%]   | 77,0505        | 94,6976        | 99,2543        | 99,8343        | 99,7514        |
| $(r)$ min [%]   | 62,3861        | 89,3952        | 97,763         | 99,2543        | 99,2543        |
| $(r)$ med [%]   | 69,1301        | 91,3007        | 98,459         | 99,4946        | 99,4946        |
| $(r)$ max [%]   | 75,145         | 94,0348        | 99,1715        | 99,6686        | 99,7514        |
| $(f)$ min [%]   | 65,6172        | 91,3007        | 98,0116        | 99,2543        | 99,2543        |
| $(f)$ med [%]   | 71,5742        | 92,6761        | 98,7655        | <b>99,5609</b> | 99,5775        |
| $(f)$ max [%]   | 76,802         | 94,6976        | 99,2543        | 99,8343        | 99,7514        |
| tempo med.      | 0,3452         | 0,4764         | 1,0721         | 2,3534         | 4,9024         |

Algumas observações podem ser feitas acerca dos resultados apresentados na Tabela 5.1:

- os resultados indicam que a utilização da norma- $\ell_1$  para classificação, utilizando uma base de treino conforme descrito na Seção 3.1, na média, apresentam uma taxa de acerto maior que a obtida pelo resíduo;
- à medida em que  $d$  aumenta, na média obtém-se maiores taxas de acerto;
- apesar de na média a taxa de acerto ter sido maior para  $d = 300$ , os mínimos e máximos para  $d = 200$  e  $300$ , quando não são iguais, são próximos (diferença de uma amostra a mais classificada erroneamente);
- o ganho, quando comparado ao custo computacional, na taxa de acerto é pequeno para  $d = 200$  e  $300$ ;
- a utilização da combinação dos três classificadores, embora não tenha apresentado a pior taxa de acerto para as dimensões  $d$  testadas, também não obteve os melhores resultados;
- embora o critério ( $u^c$ ), onde é imposta uma componente comum não nula, apresente taxas de acerto maiores quando comparadas com o critério utilizando os resíduos ( $r$ ), tal imposição não apresentou melhores resultados quando comparado com o critério ( $u$ ) sem tal imposição (exceto para  $d = 120$  e  $d = 200$ ). Isto indica que as matrizes de posto pequeno, ainda carregam informações relevantes referentes às respectivas classes;
- ao comparar os resultados obtidos com os apresentados em [56] e indicados na Tabela 5.2, exceto para dimensão  $d = 30$ , as taxas de acerto na classificação obtidas foram superiores nos testes sem oclusões.

Tabela 5.2: Resultados na taxa de reconhecimento sem oclusão. Retirado de [56].

| dimensão $d$               | 30    | 56     | 120    | 504    |
|----------------------------|-------|--------|--------|--------|
| Taxa de reconhecimento (%) | 82,6% | 91,47% | 95,53% | 98,09% |

## 5.2 Imagens de teste com oclusão

Em alguns casos, a face de teste a ser reconhecida pode apresentar oclusão parcial. Nesta seção mostra-se os testes realizados utilizando oclusão de 20%. As oclusões feitas

podem ser observadas na Figura 5.1.



Figura 5.1: Exemplo das oclusões realizadas nas imagens de teste.

Foram três localizações escolhidas para realizar a oclusão: olhos, nariz e boca. O procedimento para realização dos testes seguem o mesmo padrão dos testes realizados anteriormente, isto é, escolhe-se aleatoriamente metade do conjunto de dados como imagens de treino e a outra metade como teste (porém adicionando agora as oclusões). Para cada realização gera-se uma nova matriz de transformação,  $R$ . A dimensão escolhida para realizar estes teste foi  $d = 300$ , uma vez que foi a dimensão que obteve as maiores taxas de acerto na classificação das imagens de face no teste da seção anterior. Os resultados podem ser observados na Tabela 5.3.

Tabela 5.3: Resultados para oclusão em bloco de 20% e  $d = 300$

| Oclusão         | Olhos        | Nariz          | Boca           |
|-----------------|--------------|----------------|----------------|
| $(u^c)$ min [%] | 78,2104      | 75,3107        | 79,9503        |
| $(u^c)$ med [%] | <b>82,85</b> | <b>83,7448</b> | <b>84,1259</b> |
| $(u^c)$ max [%] | 88,3181      | 89,8094        | 89,0638        |
| $(u)$ min [%]   | 77,879       | 74,9793        | 74,6479        |
| $(u)$ med [%]   | 82,2867      | 81,9138        | 82,6429        |
| $(u)$ max [%]   | 87,0754      | 88,3181        | 88,3181        |
| $(r)$ min [%]   | 70,6711      | 70,6711        | 68,8484        |
| $(r)$ med [%]   | 76,1475      | 82,6429        | 81,9304        |
| $(r)$ max [%]   | 82,7672      | 88,3181        | 90,2237        |
| $(f)$ min [%]   | 78,1276      | 75,4764        | 77,3819        |
| $(f)$ med [%]   | 82,7341      | 83,7365        | 83,7448        |
| $(f)$ max [%]   | 88,1524      | 89,9751        | 88,9809        |
| tempo med.      | 5,0042       | 4,7212         | 5,6271         |

Pode-se observar que os melhores resultados nos testes com oclusão (destacados na tabela) foram obtidos utilizando a minimização da norma- $\ell_1$  com restrição da matriz comum diferente de zero. Novamente, a utilização da combinação dos três classificadores, embora não tenha apresentado a pior taxa de acerto para a dimensão  $d = 300$ , também não obteve os melhores resultados.

No trabalho de Wright et al. [54], o problema de oclusão é tratado de maneira diferenciada, utilizando o problema de minimização- $\ell_1$  estendido apresentado na Seção 2.3.4:

$$(\ell_1^e) : \quad \hat{\mathbf{w}}_1 = \arg \min \|\mathbf{w}\|_1 \quad \text{sujeito a} \quad \mathbf{y} = B\mathbf{w} \quad (5.1)$$

onde  $\mathbf{w} = [\mathbf{x} \quad \mathbf{e}]^T$ ,  $B = [A \quad I]$  e o modelo linear considerado é:

$$\mathbf{y} = \mathbf{y}_0 + \mathbf{e}_0 = A\mathbf{x}_0 + \mathbf{e}_0 \quad (5.2)$$

com  $\mathbf{e}$  sendo o vetor de erros/occlusão.

Observe que a matriz de  $B$  não é alterada por  $R$  como anteriormente. De fato, os resultados apresentados por Wright et al. [54] utilizam um redimensionamento das imagens (não necessário, porém realizado para executar os experimentos dentro da memória do MATLAB<sup>®</sup>) para  $96 \times 84$  *pixels*, ao invés de utilizar *randomfaces*. Além disso, a base de treino é cuidadosamente escolhida de forma que possua apenas imagens com iluminação normal para moderada<sup>1</sup>.

Apesar das taxas de acerto mostradas na Tabela 5.3 serem inferiores aos resultados obtidos no teste de oclusão por Wright et al. [54], tais resultados dependem menos do conjunto de treino escolhido e a extração de características é realizada utilizando uma matriz  $R$  gaussiana que permite reduzir a dimensão para  $d = 300$  ao invés de  $d = 8064$  pelo redimensionamento feito no trabalho de Wright et al. [54] e isto tem um impacto direto no tempo de classificação.

Até o momento, apresentou-se apenas oclusões em blocos. Entretanto, as imagens de teste podem apresentar outros tipos de oclusões. Wright et al. [54], utilizando a versão estendida, realizam testes em imagens que possuem pontos aleatoriamente corrompidos.

A Figura 5.2 mostra exemplos da oclusão aleatória na imagem de face. Para gerar as imagens de teste com tais oclusões o seguinte procedimento é adotado: escolhe-se aleatoriamente *pixels* (25%, 50% e 75% do total de *pixels*) da imagem e substitui-se os

---

<sup>1</sup>Iluminação normal foi denominada para imagens de face que possuem uma iluminação regular onde é possível visualizar claramente a face. Iluminação moderada designa-se à imagens de face que possuem uma iluminação irregular, contudo não comprometem a visualização da face

respectivos níveis de cinza por amostras independentes e identicamente distribuídas de uma distribuição uniforme no intervalo de  $[0, 255]$ .

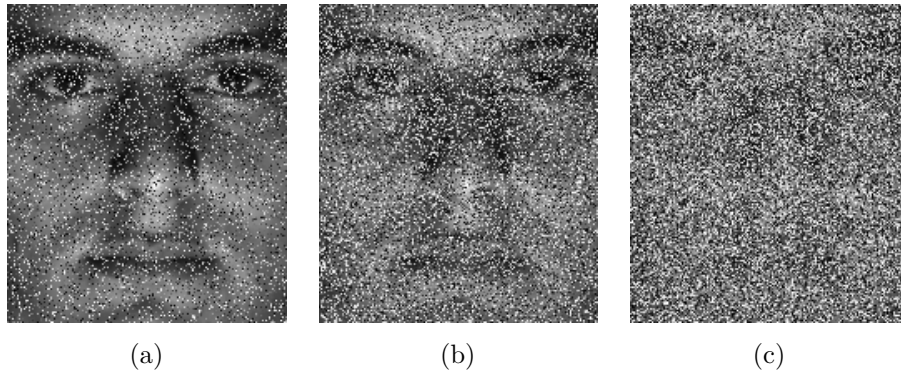


Figura 5.2: Exemplos de imagens de face com *pixels* aleatórios com níveis de cinza corrompidos. Taxa de oclusão: (a) 25%, (b) 50% e (c) 75%.

Realizou-se os mesmos testes anteriores com oclusão, entretanto, substituindo a oclusão em blocos por *pixels* aleatórios com níveis de cinza corrompidos. Para este tipo de oclusão, os resultados foram insatisfatórios (taxa de acerto inferior à 50% para uma oclusão aleatória de 25% dos *pixels* da imagem).

### 5.3 Imagens de teste com oclusão utilizando a versão estendida

Motivados pela taxa de acerto insatisfatória obtida no teste anterior, utilizou-se a versão estendida proposta por Wright et al. [54] para oclusões aleatórias, mas utilizando a norma- $\ell_1$  da função característica para determinação das classes ao invés dos resíduos. Isso é feito utilizando o Algoritmo 5. No intuito de comparar com os resultados de Wright et al. [54] e tornar possível a execução dos testes dentro da memória do MATLAB<sup>®</sup>, foi necessário realizar a redução de dimensão da imagem para  $96 \times 84$  *pixels*.

#### 5.3.1 Oclusão aleatória

Para gerar os resultados apresentados na Tabela 5.4, os seguintes passos foram realizados: escolhe-se aleatoriamente metade do conjunto de dados como imagens de treino. Das imagens restantes, escolhe-se aleatoriamente dez imagens, por classe, como

teste. Para cada imagem de teste, uma oclusão aleatória é gerada. São feitas cinco realizações e para cada uma, diferentes percentagens de oclusão foram testadas (sem oclusão, 25%, 50% e 75% de oclusão).

Tabela 5.4: Oclusão Aleatória

| % Oclusão       | 0              | 25             | 50             | 75             |
|-----------------|----------------|----------------|----------------|----------------|
| $(u^c)$ min [%] | 99,2105        | 98,9474        | 90,7895        | 24,4737        |
| $(u^c)$ med [%] | <b>99,5789</b> | <b>99,4737</b> | <b>93,5789</b> | <b>40,8421</b> |
| $(u^c)$ max [%] | 100,0000       | 100,0000       | 96,5789        | 58,4211        |
| $(u)$ min [%]   | 99,2105        | 98,9474        | 90,7895        | 24,2105        |
| $(u)$ med [%]   | <b>99,5789</b> | <b>99,4737</b> | 93,4737        | 40,7895        |
| $(u)$ max [%]   | 100,0000       | 100,0000       | 96,5789        | 58,6842        |
| $(r)$ min [%]   | 98,9474        | 98,9474        | 88,4211        | 20,5263        |
| $(r)$ med [%]   | 99,5263        | <b>99,4737</b> | 91,2105        | 35,8421        |
| $(r)$ max [%]   | 100,0000       | 100,0000       | 94,4737        | 53,4211        |
| $(f)$ min [%]   | 99,2105        | 98,9474        | 90,7895        | 24,2105        |
| $(f)$ med [%]   | <b>99,5789</b> | <b>99,4737</b> | <b>93,5789</b> | 40,7368        |
| $(f)$ max [%]   | 100,0000       | 100,0000       | 96,5789        | 58,4211        |

Devido a alta dimensão dos dados (agora  $d = 8064$ ), a otimização para um conjunto de treino grande se torna ainda mais custosa computacionalmente. Portanto, ao invés de utilizar um número igual à metade de faces disponível na base de treino para agrupamento, utilizou-se apenas quatro agrupamentos na etapa 4 do Algoritmo 3. Logo, cada classe é composta por cinco amostras de treino (a imagem comum mais as quatro oriundas do agrupamento das matrizes de posto pequeno).

Os resultados mostrados na Tabela 5.4 indicam que para uma taxa de oclusão de até 25%, os critérios apresentam resultados semelhantes. Para oclusões de 50% ou mais os critérios utilizando a norma- $\ell_1$  apresentaram uma taxa de acerto, na média, maior. A Figura 5.2c mostra uma imagem de face com 75% de oclusão. Pode-se observar que a face já é irreconhecível para o ser humano e o sistema de classificação também não é capaz de classificar corretamente as faces. Esse mesmo comportamento é observado nos teste realizados em Wright et al. [54]. Para taxas de oclusão acima de 70%, a taxa de reconhecimento caiu significativamente conforme pode ser observado na Tabela 5.5. Novamente, ressalta-se que os resultados apresentados na Tabela 5.4 não se baseia em uma escolha específica da base de treino como apresentado em [54] e indicados na Tabela 5.5.

Tabela 5.5: Resultados na taxa de reconhecimento com oclusões aleatórias nas imagens de teste. Retirado de [54].

| % de pixels corrompidos | 0%   | 10%  | 20%  | 30%  | 40%  | 50%  | 60%   | 70%   | 80%   | 90%  |
|-------------------------|------|------|------|------|------|------|-------|-------|-------|------|
| Taxa de reconhecimento  | 100% | 100% | 100% | 100% | 100% | 100% | 99,3% | 90,7% | 37,5% | 7,1% |

Vale ressaltar também, que os autores de [54] não mencionam quantas realizações foram feitas e nem se o resultado apresentado é o melhor ou o pior entre todas as realizações. Observa-se na Tabela 5.4 que os melhores resultado para até 25% de oclusão são 100% na taxa de acerto. Entretanto, na média, tal valor não se manteve, devido às diferentes realizações na escolha da base de treino. Outro ponto que deve ser mencionado neste momento é que nos resultados apresentados na Tabela 5.4, o número de elementos na base de treino diminuiu consideravelmente. Mesmo assim, os resultados apresentados estão coerentes aos observados por Wright et al. [54], sugerindo pouco impacto dessa etapa na taxa de acerto e um ganho no tempo de processamento.

Outra observação a ser feita é em relação ao tempo de processamento. O tempo médio para classificação de uma face foi de 41 segundos. Isto indica que o custo computacional para a solução do problema de otimização estendido é mais custoso. A partir deste momento, deixa-se de apresentar os tempos médios nas tabelas de resultados, já que o intuito era apresentar uma ideia de tempo para classificação de imagens de teste, em especial, comparar o tempo médio para classificar uma imagem usando o Algoritmo 4 e Algoritmo 5.

### 5.3.2 Oclusão em blocos em posição aleatória e imagem não correlacionada

Para os testes apresentados nesta seção, utilizou-se uma imagem não face humana (no caso, a imagem utilizada foi a de um babuíno, comumente utilizada em testes de processamento de imagens) como oclusão. A posição de tal oclusão é feita de maneira aleatória com as seguintes percentagens 10%, 25% e 50% de oclusão. As imagens da Figura 5.3 mostram exemplos de tais oclusão.

Para avaliação numérica, utiliza-se os procedimentos adotados anteriormente. São utilizadas dez imagens de teste de cada classe, escolhidas aleatoriamente, e a solução é dada pela versão estendida. Os resultados podem ser observados na Tabela 5.6.

Os resultados apresentados na Tabela 5.6 indicam que os melhores resultados ainda são dados pela utilização do critério baseado na norma- $\ell_1$ , exceto para o caso de 25%

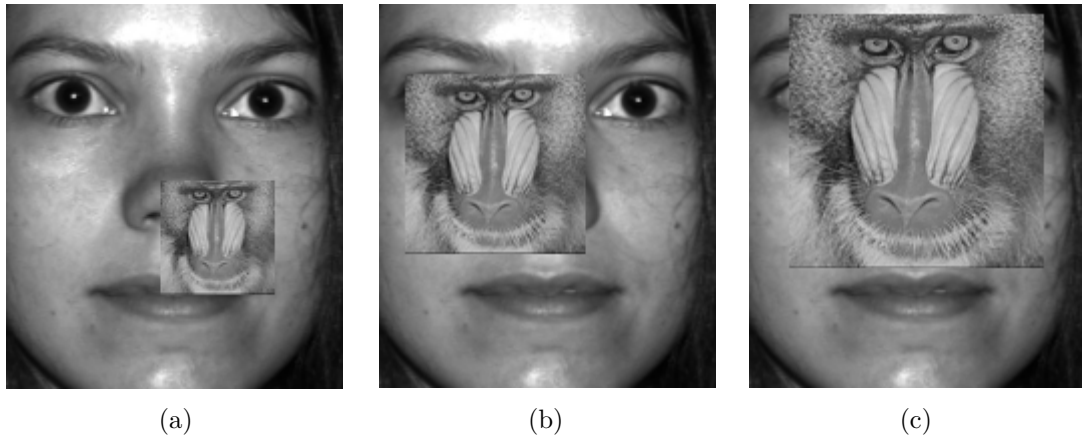


Figura 5.3: Exemplos de imagens de face com oclusão em blocos, em posição aleatória, utilizando imagem não relacionada. Taxa de oclusão: (a) 10%, (b) 25% e (c) 50%.

Tabela 5.6: Oclusão em blocos (usando imagem não correlacionada) em posição aleatória para diferentes taxas de oclusão

| % Oclusão       | 10             | 25             | 50             |
|-----------------|----------------|----------------|----------------|
| $(u^c)$ min [%] | 97,1053        | 86,8421        | 45,2632        |
| $(u^c)$ med [%] | <b>98,3158</b> | 88,3158        | <b>49,2632</b> |
| $(u^c)$ max [%] | 98,9474        | 90,5263        | 53,9474        |
| $(u)$ min [%]   | 97,1053        | 86,0526        | 42,6316        |
| $(u)$ med [%]   | <b>98,3158</b> | 87,2105        | 47,5789        |
| $(u)$ max [%]   | 98,9474        | 88,9474        | 53,1579        |
| $(r)$ min [%]   | 96,8421        | 86,8421        | 43,4211        |
| $(r)$ med [%]   | 97,8947        | <b>88,4211</b> | 47,6842        |
| $(r)$ max [%]   | 98,6842        | 89,2105        | 53,6842        |
| $(f)$ min [%]   | 97,1053        | 86,8421        | 44,2105        |
| $(f)$ med [%]   | 98,2632        | 88,3158        | 48,8421        |
| $(f)$ max [%]   | 98,9474        | 90,5263        | 54,2105        |

de oclusão, onde a norma- $\ell_2$  teve a melhor taxa de classificação. Outra observação importante a se fazer neste momento é que tais resultados indicam que este tipo de oclusão foi consideravelmente mais danosa às imagens para a classificação do que a oclusão aleatória (*pixels* aleatoriamente escolhidos e corrompidos). Isso já era esperado, já que os teste de oclusões mostrados na Seção 4.4 indicam que oclusões aleatórias são melhores estimadas, devido sua natureza esparsa, do que oclusões onde o conteúdo de



tais oclusões não são imagens face humana.

Tabela 5.7: Resultados na taxa de reconhecimento com oclusões em blocos utilizando imagens não correlacionadas. Retirado de [54].

| % de pixels corrompidos | 0%   | 10%  | 20%   | 30%   | 40%   | 50%   |
|-------------------------|------|------|-------|-------|-------|-------|
| Taxa de reconhecimento  | 100% | 100% | 99,8% | 98,5% | 90,3% | 65,3% |

Os resultados exibidos na Tabela 5.6 indicam que as taxas de classificação para oclusões em blocos são inferiores quando comparadas com as oclusões aleatórias apresentadas na Tabela 5.4. Isto já era esperado, em vista dos testes feitos no capítulo anterior, onde foi possível observar que para oclusões em blocos a solução encontrada era menos esparsa.

Ao comparar os resultados da Tabela 5.6 com os resultados apresentados por Wright et al. [54], indicados na Tabela 5.7, as taxas de classificação sugerem que o sistema proposto possui uma taxa de classificação inferior nas condições simuladas.

Além da base de treino ser cuidadosamente escolhida em [54], outro motivo que contribuiu para esses resultados refere-se à drástica redução na base de treino. No caso onde as imagens apresentam oclusões, a redução da base de treino afetou consideravelmente os resultados na classificação das imagens de face. Para demonstrar este efeito, realizou-se o mesmo experimento anterior, contudo utilizando dez agrupamentos ao invés dos quatro grupos. Os resultados são apresentados na Tabela 5.8. É possível observar uma melhora na taxa de acerto de classificação das faces com oclusão, compatível ainda com a apresentada por Wright et al. [54] com a ressalva de que nenhuma escolha diferenciada é feita para composição da base de treino e teste nos resultados da Tabela 5.8.

## 5.4 Oclusão na base de treino

Os testes apresentados até agora apenas levam em consideração oclusões nas imagens de teste. Isso porque a base de treino, exceto pela variação de iluminação, é bem controlada. Entretanto, há casos onde as imagens disponíveis para compor a base de treino possuem oclusão/ruído. Tais oclusões podem comprometer o classificador, uma vez que podem mascarar informações relevantes presentes na imagem. Isso fica mais evidente em sistemas baseados na geometria da face (como distância entre olhos, orelha

Tabela 5.8: Oclusão em blocos (usando imagem não correlacionada) em posição aleatória para diferentes taxas de oclusão usando 10 *clusters* por classe

| % Oclusão       | 10             | 25             | 50             |
|-----------------|----------------|----------------|----------------|
| $(u^c)$ min [%] | 98,9474        | 95,2632        | 61,5789        |
| $(u^c)$ med [%] | <b>99,5263</b> | <b>96,0000</b> | <b>65,5263</b> |
| $(u^c)$ max [%] | 99,7368        | 97,6316        | 69,2105        |
| $(u)$ min [%]   | 98,9474        | 94,4737        | 57,6316        |
| $(u)$ med [%]   | 99,4737        | 95,5789        | 62,4737        |
| $(u)$ max [%]   | 99,7368        | 97,3684        | 66,5789        |
| $(r)$ min [%]   | 98,9474        | 94,4737        | 60,7895        |
| $(r)$ med [%]   | 99,3684        | 95,9474        | 63,3158        |
| $(r)$ max [%]   | 100,0000       | 97,1053        | 66,3158        |
| $(f)$ min [%]   | 98,9474        | 95,0000        | 60,2632        |
| $(f)$ med [%]   | 99,4737        | 95,8947        | 64,6842        |
| $(f)$ max [%]   | 99,7368        | 97,6316        | 68,9474        |

e boca) como extração de características para a composição da base de treino. Se essas estiverem corrompidas, qualquer imagem de teste, mesmo que sem oclusões/ruídos, pode ser classificada erroneamente. Para o caso do classificador baseado em representação esparsa, isso não é diferente. Uma vez que tenta-se representar (de maneira esparsa) as imagens de teste a partir das imagens de treino, é de se esperar que se as imagens de treino tiverem oclusões/ruído, tal representação fique comprometida. Nesta seção avalia-se o desempenho do sistema mesmo que este tenha como restrição trabalhar com oclusões em imagens de treino. Vale ressaltar que nenhuma medida diferenciada é feita para tratar o problema, mas acredita-se que com a metodologia proposta para a composição da base de treino, o sistema será capaz de isolar as oclusões/ruídos, ou parte delas.

A Figura 5.4 mostra exemplos de faces do conjunto de treino, de uma classe, com diferentes iluminações. Como descrito na Seção 3.1, aplica-se a equalização de histograma nas imagens e o resultado pode ser observado na Figura 5.5a. A Figura 5.5b mostra o resultado da equalização de histograma para as mesmas imagens, contudo, com a adição de uma oclusão aleatória de 10%. Para oclusões de 25% e 50%, os resultados são mostrados na Figura 5.5c e Figura 5.5d, respectivamente.

A utilização do modelo JSM-MC mostrou-se capaz de concentrar as informações esparsas, incluindo as deteriorações resultantes da equalização de histograma, em uma

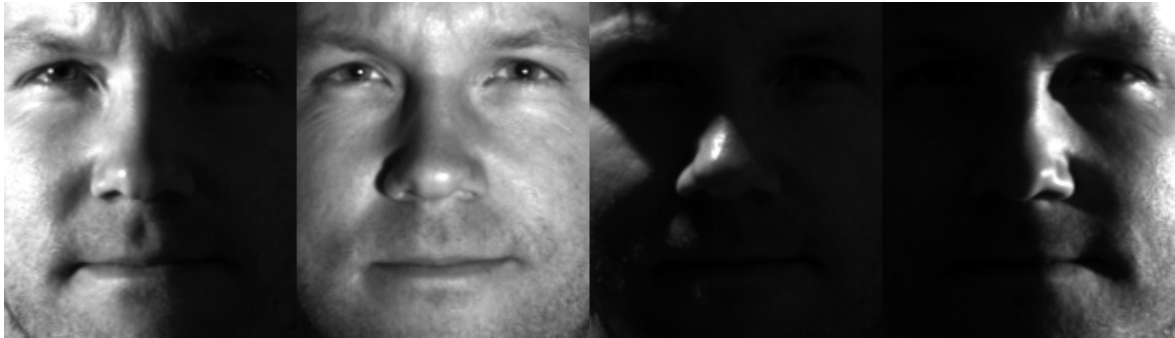


Figura 5.4: Exemplos de imagens de treino da mesma classe

matriz que não é utilizada para compor a base de treino. Contudo, além da deterioração indesejada oriunda da equalização de histograma, as imagens de treino apresentam as oclusões aleatórias inseridas em todas as imagens.

Ao aplicar o modelo JSM-MC nas imagens com oclusões, é possível observar que a imagem comum para 10% de oclusão (Figura 5.6b) apresenta poucas diferenças quando comparada com a imagem comum da decomposição JSM-MC das faces sem oclusão (Figura 5.6a). Para 25% de oclusão, a imagem comum começa a apresentar diferenças mais perceptíveis ao redor dos olhos e nariz (Figura 5.6c) e para 50% a imagem comum (Figura 5.6d) passa a ser afetada pela oclusão aleatória das imagens de entrada. O mesmo pode ser observado para as imagens correspondentes às matrizes de posto pequeno, conforme mostra a Figura 5.7.

Para avaliar numericamente a semelhança entre a decomposição usando o JSM-MC das imagens com oclusão e sem oclusão do teste anterior, as PSNRs das imagens comuns e de posto pequeno, oriundas da decomposição das imagens com oclusão, em relação às respectivas imagens da decomposição usando imagens sem oclusão foram calculadas e podem ser observadas na Tabela 5.9. Os valores de PSNRs apresentados na Tabela 5.9 confirmam as observações feitas anteriormente.

Tabela 5.9: PSNR das componentes comum e de posto pequeno para imagens de treino com oclusão aleatória das 4 imagens mostradas na Figura 5.4

| % de oclusão                        | 10      | 25      | 50      |
|-------------------------------------|---------|---------|---------|
| Componente comum                    | 40,7330 | 31,8669 | 21,1621 |
| Matriz de posto pequeno da imagem 1 | 35,7764 | 27,6542 | 15,3098 |
| Matriz de posto pequeno da imagem 2 | 35,7763 | 28,0739 | 17,3877 |
| Matriz de posto pequeno da imagem 3 | 34,2769 | 26,5955 | 17,6342 |
| Matriz de posto pequeno da imagem 4 | 31,1794 | 23,5024 | 14,4840 |



(a)



(b)



(c)



(d)

Figura 5.5: Exemplos de imagens de treino da mesma classe após equalização de histograma para imagens (a) sem oclusão, (b) com 10%, (c) 25% e (d) 50% de oclusão aleatória.

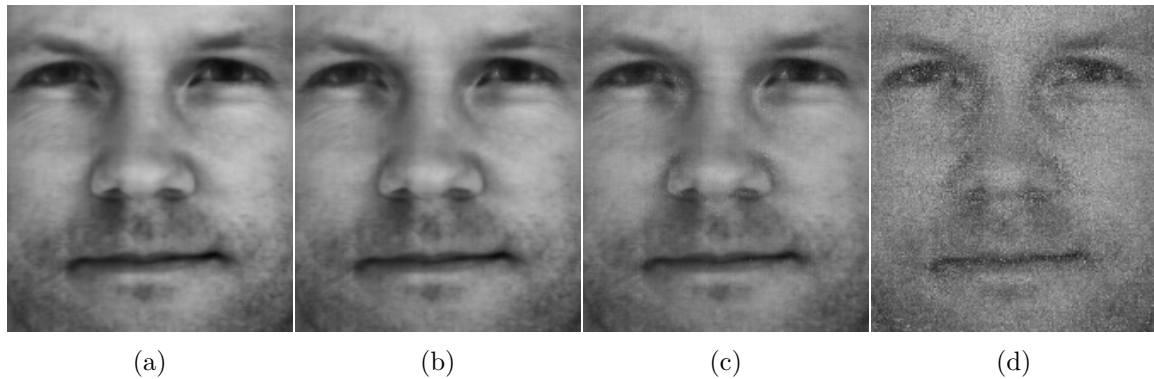


Figura 5.6: Imagem comum da decomposição JSM-MC (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão.

Como justificado anteriormente, não se utiliza as matrizes esparsas para a composição da base de treino. A Figura 5.8 ratifica esta escolha, uma vez que grande parte da oclusão aleatória é representada por essas matrizes. Isso torna-se ainda mais evidente ao realizar a reconstrução das imagens de face utilizando apenas a matriz comum e as matrizes de posto pequeno. Neste caso, conforme mostrado na Figura 5.9 é possível observar uma melhora significativa nas imagens de face reconstruídas quando comparadas com as imagens após equalização de histograma e inserção da oclusão aleatória (Figura 5.5).

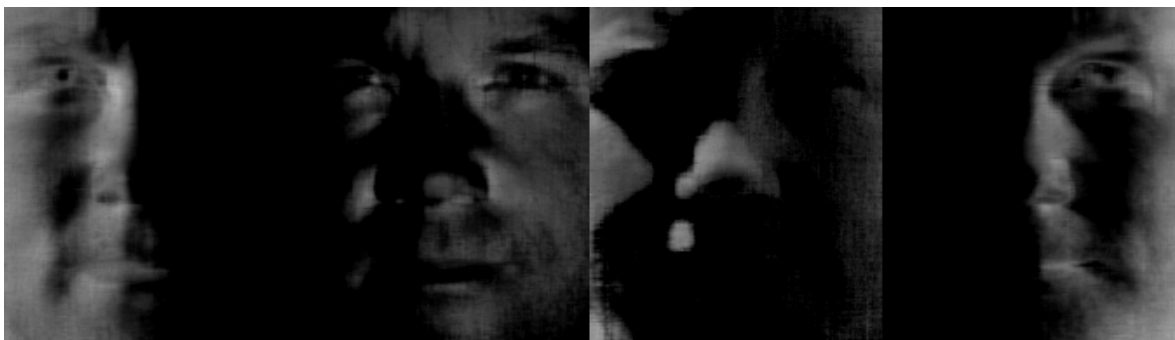
As PSNRs foram calculadas para avaliar numericamente a influência da matriz esparsa  $E$ . Neste caso, para cada uma das quatro imagens, foi calculada a PSNR das imagens reconstruídas com e sem a matriz esparsa  $E$ . Isso foi feito para 10%, 25% e 50% de oclusão nas imagens. Os resultados podem ser observados na Tabela 5.10. Para todos os casos, é possível observar um melhora no PSNR quando as matrizes esparsas não são utilizadas.

Tabela 5.10: PSNRs das quatro imagens reconstruídas com e sem a matriz esparsa  $E$  para 10%, 25% e 50% de oclusão

|          | 10%     |         | 25%     |         | 50%     |         |
|----------|---------|---------|---------|---------|---------|---------|
|          | Sem $E$ | Com $E$ | Sem $E$ | Com $E$ | Sem $E$ | Com $E$ |
| Imagem 1 | 30,4829 | 17,7378 | 24,8078 | 13,6139 | 15,3299 | 10,6822 |
| Imagem 2 | 29,0600 | 16,3618 | 24,7960 | 12,6226 | 15,3188 | 9,89680 |
| Imagem 3 | 28,5461 | 16,2492 | 24,2710 | 12,5698 | 15,2767 | 9,90480 |
| Imagem 4 | 26,4776 | 16,4741 | 20,6755 | 12,7078 | 13,0852 | 10,0366 |



(a)



(b)



(c)

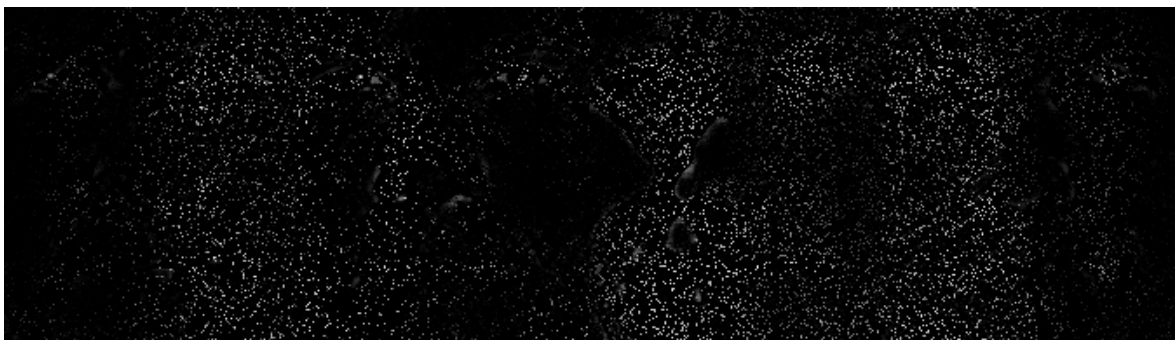


(d)

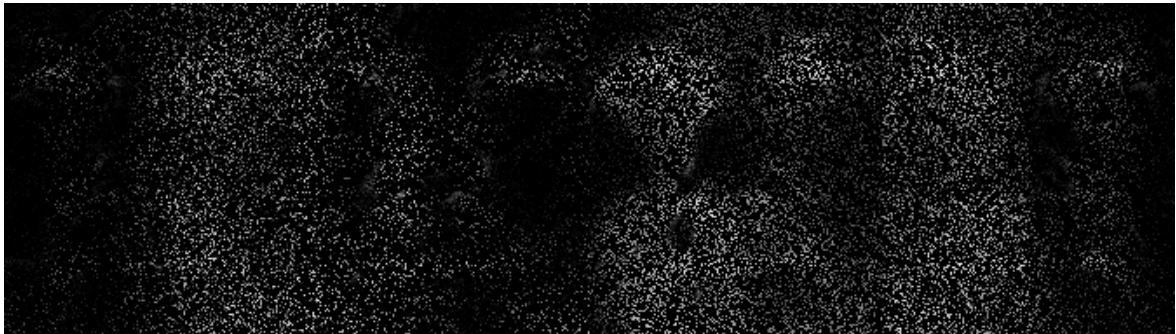
Figura 5.7: Imagens referentes às 4 matrizes de posto pequeno da decomposição JSM-MC (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão.



(a)



(b)



(c)



(d)

Figura 5.8: Imagens referentes às 4 matrizes esparsas da decomposição JSM-MC (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão.



(a)



(b)



(c)



(d)

Figura 5.9: Imagens reconstruídas utilizando apenas a matriz comum e as matrizes de posto pequeno para as imagens de treino (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão aleatória.



A última etapa da composição da base de treino, consiste no agrupamento. Para efeito de visualização, utilizou-se quatro grupos no agrupamento. Os resultados podem ser observados na Figura 5.10. Como é obtido o centróide para os grupos nesse procedimento, o efeito da oclusão aleatória pode ser ainda mais atenuado. As PSNRs podem ser observadas na Tabela 5.11.

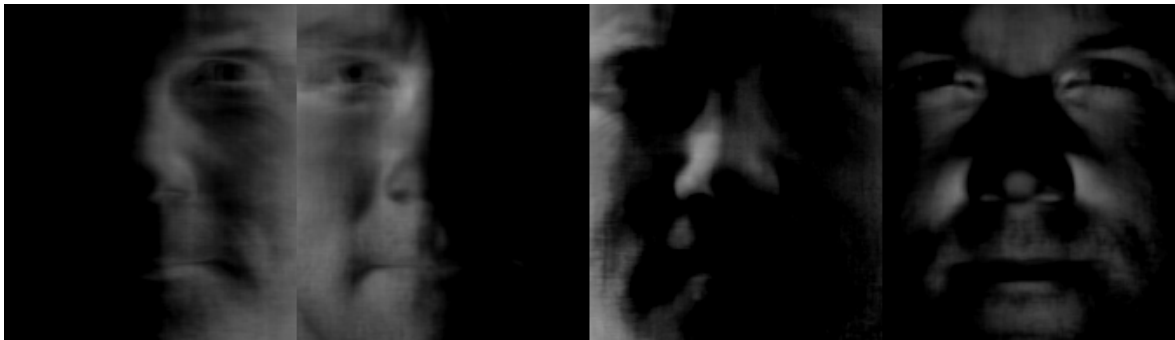
Tabela 5.11: PSNRs dos centróides resultantes do agrupamento das componentes de posto pequeno, para imagens de treino com oclusão aleatória, mostrados na Figura 5.10

| % de oclusão | 10      | 25      | 50      |
|--------------|---------|---------|---------|
| Centróide 1  | 35,7911 | 27,9842 | 21,6848 |
| Centróide 2  | 40,6310 | 32,4776 | 22,5698 |
| Centróide 3  | 30,7078 | 22,9214 | 16,8894 |
| Centróide 4  | 38,4869 | 30,3491 | 22,3567 |

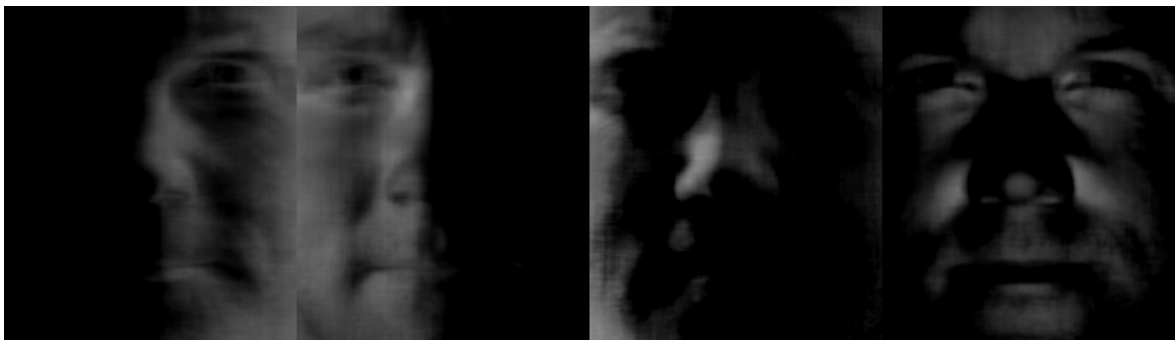
Por fim, realiza-se a classificação de imagens de teste sem oclusão, contudo, em todas as imagens de treino foram adicionadas oclusões aleatórias. Isto é feito para isolar apenas o efeito das oclusões em imagens de treino e assim avaliar a capacidade do sistema de compensar tais oclusões.

Como nos testes anteriores, a imagens de treino (metade da base de dados) são escolhidas aleatoriamente. Os testes são realizados com taxas de 10%, 25% e 50% de oclusão aleatória nas imagens de treino. Para cada taxa, dez realizações são feitas. Nestes testes, a versão estendida não é utilizada, e portanto, para cada realização uma matriz de transformação  $R$  diferente é gerada, conforme Definição 2.1, e a dimensão  $d = 300$  foi usada. Os resultados, para estes testes, são apresentados na Tabela 5.12.

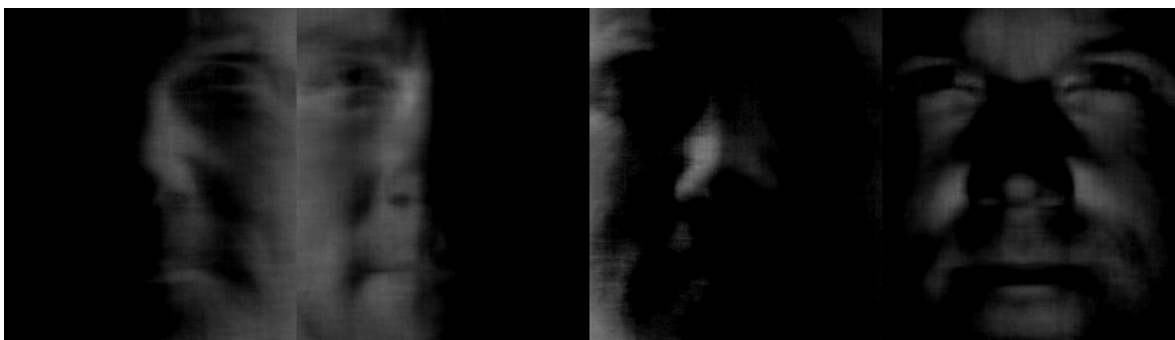
Os testes envolvendo a combinação dos três classificadores não foram realizadas, visto os resultados apresentados anteriormente, em que apesar de não apresentarem as piores taxas de acertos, também não apresentavam as melhores taxas. Ao comparar estes resultados com a Tabela 5.1 para  $d = 300$ , é possível verificar que para taxas até 25% de oclusão, os resultados com oclusão aleatória na base de treino é similar aos resultados para o caso onde não há oclusão. Para o caso de oclusão de 50%, os resultados apresentam uma taxa de acerto menor, mas satisfatória, visto a taxa de oclusão aplicada.



(a)



(b)



(c)



(d)

Figura 5.10: Imagens referentes ao agrupamento realizado nas matrizes de posto pequeno da decomposição JSM-MC das imagens de treino (a) sem oclusão, (b) 10%, (c) 25% e (d) 50% de oclusão aleatória.

Tabela 5.12: Testes utilizando oclusões aleatórias de 10%, 25% e 50% na base de treino

| % Oclusão       | 10             | 25             | 50             |
|-----------------|----------------|----------------|----------------|
| $(u^c)$ min [%] | 99,5857        | 99,3372        | 88,9809        |
| $(u^c)$ med [%] | <b>99,6686</b> | <b>99,5775</b> | <b>90,6711</b> |
| $(u^c)$ max [%] | 99,7514        | 99,7514        | 92,2121        |
| $(u)$ min [%]   | 99,5857        | 99,3372        | 88,9809        |
| $(u)$ med [%]   | 99,6437        | 99,5692        | 90,6628        |
| $(u)$ max [%]   | 99,7514        | 99,7514        | 92,2121        |
| $(r)$ min [%]   | 99,3372        | 99,0058        | 87,0754        |
| $(r)$ med [%]   | 99,5775        | 99,4532        | 89,5775        |
| $(r)$ max [%]   | 99,8343        | 99,7514        | 91,3007        |

## 5.5 Rejeição de imagens de teste

A rejeição de imagens de teste é uma etapa importante e deve ser adaptada para cada aplicação. Em aplicações de extrema segurança onde não se pode tolerar a entrada de intrusos, o sistema deve ser capaz de recusar imagens associadas à múltiplas classes, mesmo que isto represente uma rejeição maior de imagens que seriam classificadas corretamente. Assim como aplicações em detecção de faces em imagens digitais, tal rigorosidade pode não ser necessária.

Esta seção verifica-se a capacidade do sistema de rejeitar imagens de teste que não pertencem à uma das classes da base de treino, ou que devido ao ruído inserido em tais imagens impossibilite a correta identificação da face.

Yang et al. [56] definiram o índice de concentração de esparsidade (SCI) como:

**Definição 5.1.** *Sparsity Concentration Index* [56]: Seja  $\mathbf{x}$  a solução esparsa de  $(\ell_1)$  (2.7) ou  $(\ell_{1_s})$  (2.4). O índice de concentração esparsa (SCI) de  $\mathbf{x}$  é definido como:

$$SCI(\mathbf{x}) = \frac{k \max_i \|\delta_i(\mathbf{x})\|_1 / \|\mathbf{x}\|_1 - 1}{k - 1} \in [0, 1]$$

onde  $k$  é o número de classes da base de dados.

Quando  $SCI(\mathbf{x}) = 1$ , a imagem de teste é representada usando apenas imagens de uma classe e quando  $SCI = 0$ , os coeficientes esparsos estão espalhados sobre todas as classes. Logo, aplica-se um limiar  $\tau \in (0, 1)$  e rejeita-se tal imagem de teste caso

$$SCI(\mathbf{x}) \leq \tau. \quad (5.3)$$

Para realizar os testes de rejeição os seguintes passos foram realizados: escolhe-se aleatoriamente metade do conjunto de dados como base de treino. Das imagens restantes, escolhe-se aleatoriamente dez imagens, por classe, como imagens de teste. Entretanto, do conjunto de treino gerado, retira-se metade das classes. Portanto, metade das imagens de teste, agora são novas para o sistema, e como não pertencem a nenhuma classe do novo conjunto de treino, devem ser rejeitadas. São feitas cinco realizações, novamente, para garantir que o resultado não seja polarizado por um conjunto específico. Para cada representação esparsa, aplica-se o índice de concentração esparsa conforme Definição 5.1. Então, aplica-se o limiar mostrado em (5.3). O limiar  $\tau$  foi escolhido de forma a obter taxas iguais de acerto (face ser classificada como face e não face como não face), e conseqüentemente iguais taxas de erro. Tal ponto é comumente referido na literatura como *equal error rate* (EER) ou taxa de erro igual. A Tabela 5.13

mostra a matriz de confusão para rejeição de faces no teste descrito. Observe que a Tabela 5.13 possui valores fracionários. Isso acontece porque é feita uma média, já que cinco realizações foram feitas.

Tabela 5.13: Matriz de Confusão para rejeição de faces sem oclusão.  $\tau = 0,2269$

|      |          | Estimada |          |       |
|------|----------|----------|----------|-------|
|      |          | Face     | Não Face | Total |
| Real | Face     | 182,6    | 7,4      | 190   |
|      | Não Face | 7,4      | 182,6    | 190   |
|      | Total    | 190      | 190      | 380   |

Outro teste realizado, ainda no contexto de rejeição de faces de teste, foi a inserção de oclusão aleatória nas imagens de teste. Portanto, além de metade das imagens de teste não possuírem uma classe no conjunto de treino, todas as imagens de teste apresentam agora oclusão. A Tabela 5.14 mostra a matriz de confusão para o teste com oclusão de 25% e a Tabela 5.15 para o teste com 50% de oclusão aleatória.

Tabela 5.14: Matriz de Confusão para rejeição de faces com 25% de Oclusão.  $\tau = 0,2405$

|      |          | Estimada |          |       |
|------|----------|----------|----------|-------|
|      |          | Face     | Não Face | Total |
| Real | Face     | 181      | 9        | 190   |
|      | Não Face | 9        | 181      | 190   |
|      | Total    | 190      | 190      | 380   |

Tabela 5.15: Matriz de Confusão para rejeição de faces com 50% de oclusão.  $\tau = 0,2356$

|      |          | Estimada |          |       |
|------|----------|----------|----------|-------|
|      |          | Face     | Não Face | Total |
| Real | Face     | 172,8    | 17,2     | 190   |
|      | Não Face | 17,2     | 172,8    | 190   |
|      | Total    | 190      | 190      | 380   |

Ao comparar os valores das Tabela 5.14 e Tabela 5.15 com a Tabela 5.13, para 25% de oclusão, pouco efeito é observado na rejeição de faces. Isto é, ao resolver o problema de otimização estendido, o erro estimado é próximo do erro real (oclusão artificial aleatória inserida) conforme verificado na Seção 5.2. Para oclusão de 50%, o efeito da oclusão possui mais impacto na rejeição de faces fazendo com que um número

maior de Não Face sejam classificados como Face. Isso se deve ao fato de metade dos dados da imagem terem sido corrompidos, dificultando assim, a estimação da oclusão na solução do problema de otimização. Em outras palavras, os coeficientes da representação esparsa ficam mais espalhados (associados à múltiplas classes) a medida que a oclusão aumenta. Vale ressaltar ainda que para a Tabela 5.14 e Tabela 5.15 utilizou-se a versão estendida (Algoritmo 5) para achar a representação esparsa das imagens de teste, e portanto, para oclusões pequenas, o erro estimado compensa parte da oclusão inserida.

Como mencionado anteriormente, a escolha de  $\tau$  depende da aplicação final do sistema. Logo, o ponto EER pode não ser o ideal, dependendo da aplicação. A Figura 5.11 mostra a matriz de confusão, com a influência de diferentes  $\tau$ , para imagens de face sem oclusão. A Figura 5.12 e Figura 5.13 mostram a influência de  $\tau$  na rejeição de imagens de teste com oclusão de 25% e 50% respectivamente. A interpretação dessas imagens segue a interpretação de uma matriz de confusão tradicional. Entretanto, ao invés de avaliar apenas um  $\tau$  específico, avalia-se a rejeição ao variar  $\tau$  entre os diferentes valores possíveis de limiar.

A determinação do limiar que resultou em rejeições iguais foi feita a partir do  $\tau$  correspondente ao cruzamento dos gráficos resultantes, após variação de  $\tau$ , das faces de teste serem reconhecidas como face e as não faces serem reconhecidas como não face. Portanto, tal limiar é dependente do conjunto de teste utilizado. Como foi possível observar, este parâmetro varia com a quantidade de oclusão que as imagens de teste apresentam. Fica claro que a determinação do  $\tau$  ideal (seja qual for a aplicação) não é uma tarefa trivial, sem o conhecimento do conjunto de teste (que é o caso da maioria das aplicações reais).

Após a variação de  $\tau$ , a curva ROC foi obtida para diferentes taxas de oclusão (sem oclusão, 25% e 50% de oclusão), conforme mostra Figura 5.14. É possível observar que a curva ROC sem oclusão e com oclusão de 25% se aproximam de uma curva ROC ideal.

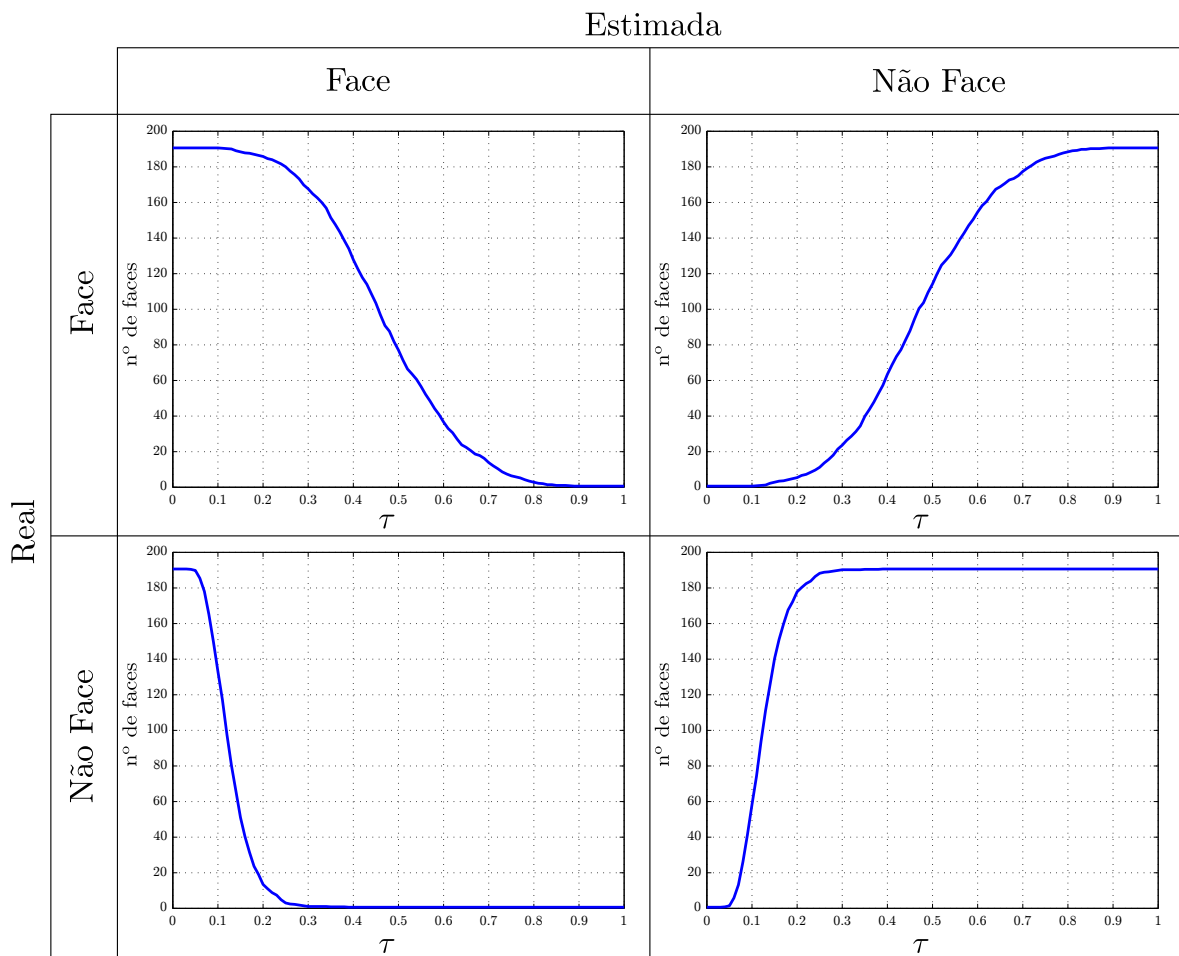


Figura 5.11: Matriz de Confusão para rejeição de faces sem oclusão variando  $\tau$

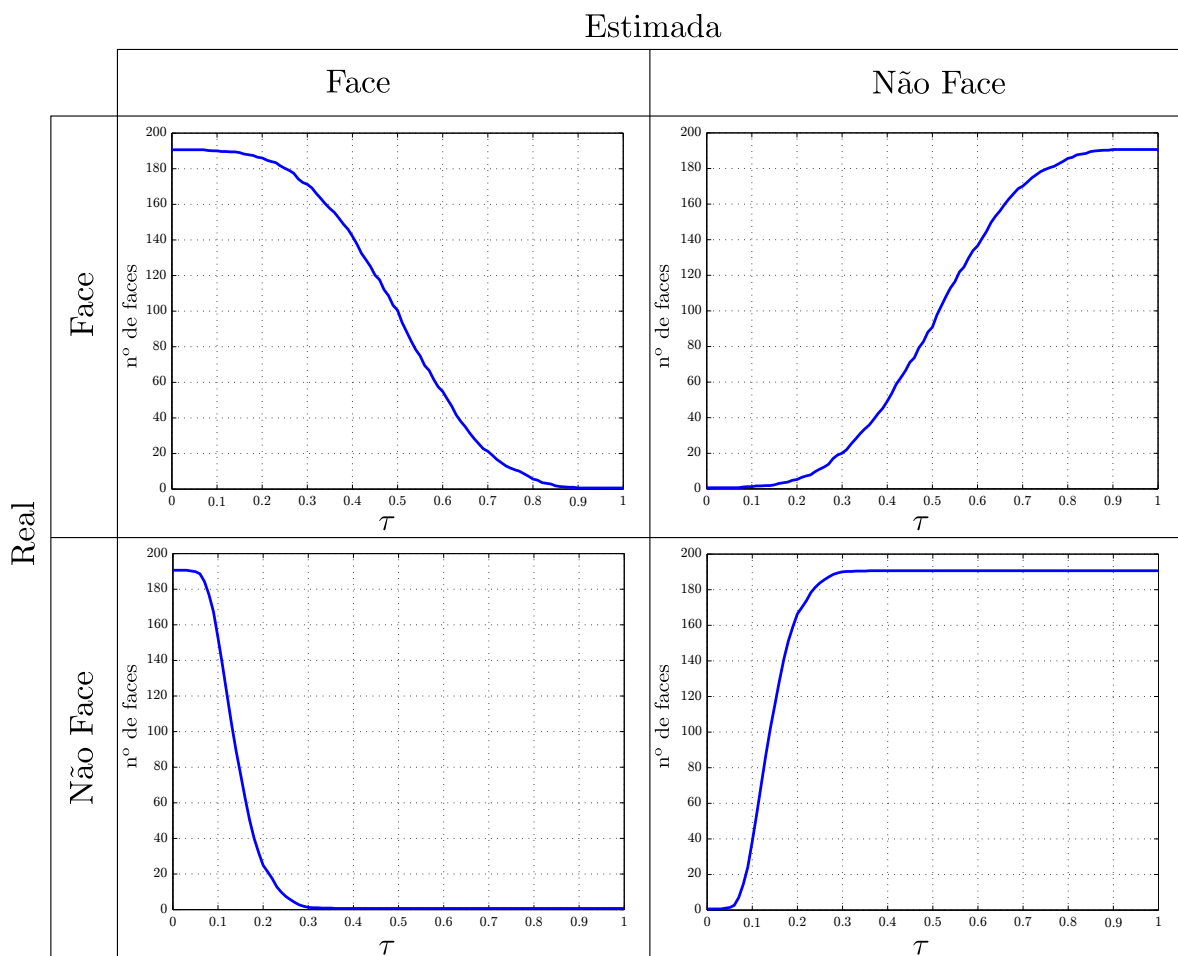


Figura 5.12: Matriz de Confusão para rejeição de faces com oclusão de 25% e variação de  $\tau$



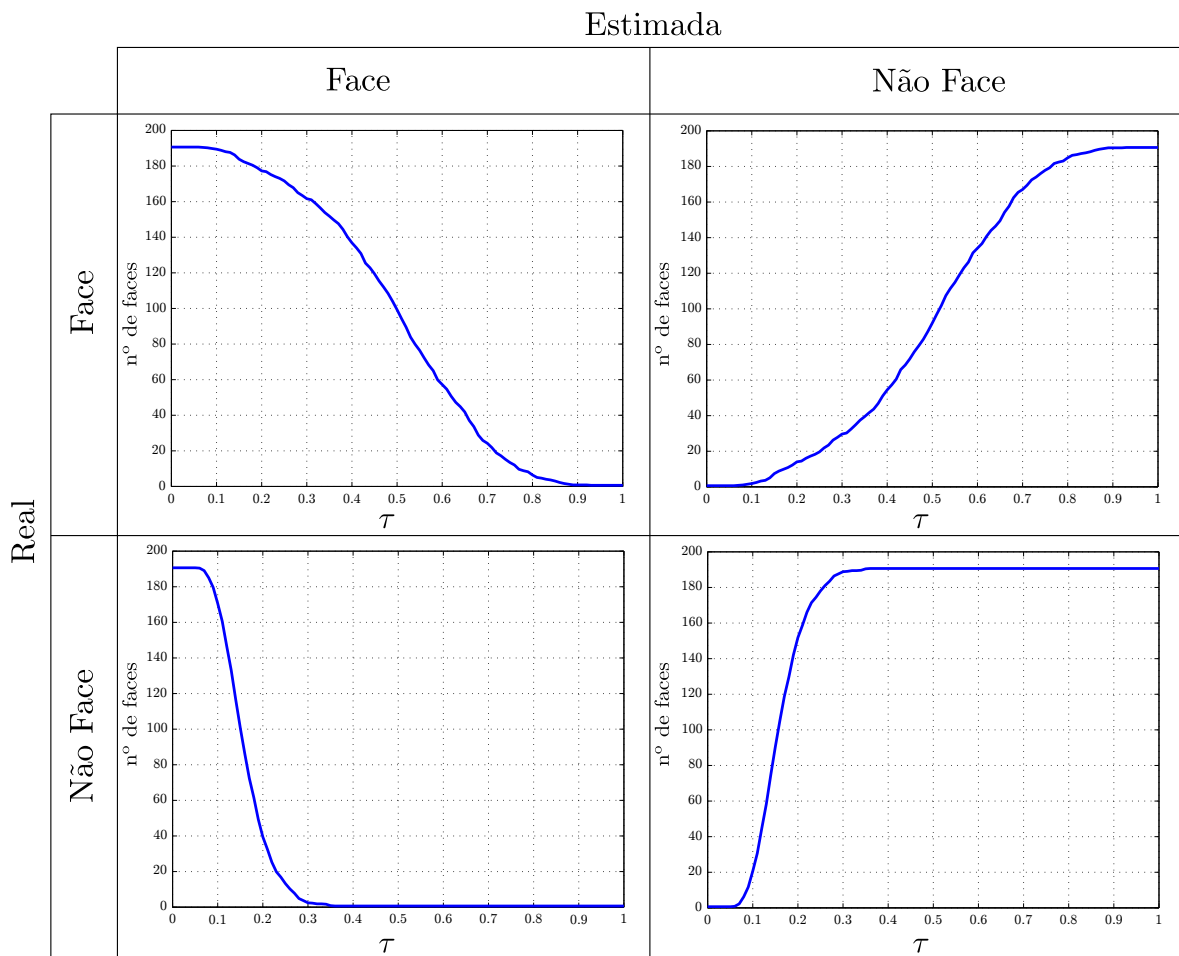


Figura 5.13: Matriz de Confusão para rejeição de faces com oclusão de 50% e variação de  $\tau$

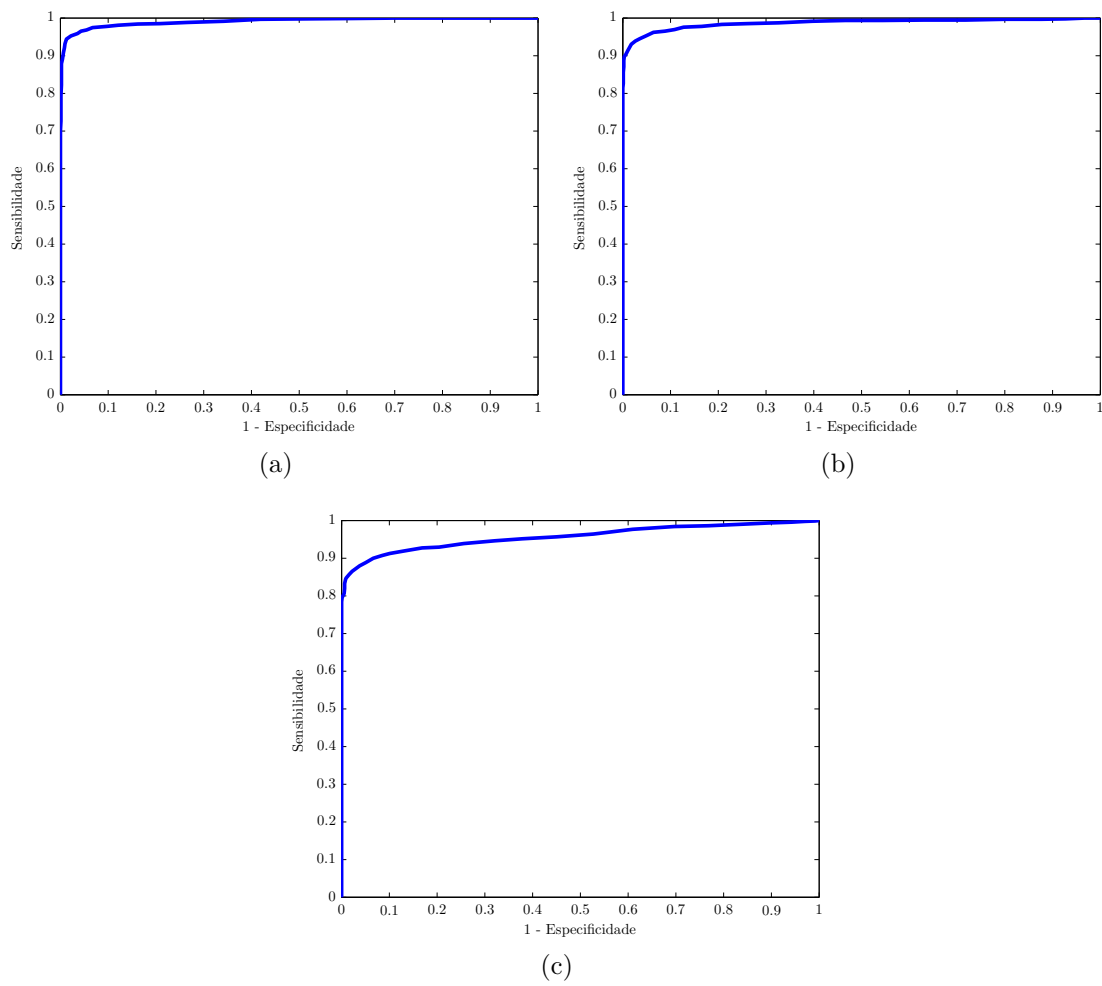


Figura 5.14: Curva ROC para rejeição de faces com (a) 0%, (b) 25% e (c) 50% de oclusão.

# Capítulo 6

## Conclusões e trabalhos futuros

Neste trabalho, o problema de reconhecimento de faces em diferentes condições de iluminação e oclusões é abordado utilizando como base o modelo de esparsidade conjunta com complementamento de matrizes para composição da base de treino e o algoritmo de classificação baseada em representação esparsa, com modificações, para a classificação das imagens de face.

Os resultados apresentados para o casos onde não há oclusão nas imagens de teste sugerem que maiores taxas de acerto, quando comparada às taxas obtidas por Wright et al. [54], podem ser alcançadas utilizando a combinação do JSM-MC para composição da base de treino e o SRC modificado para a classificação. Além disso, no caso de oclusões, a versão estendida mostrou-se superior quando comparado com o Algoritmo 4 uma vez que a oclusão já é considerada no modelo do problema de otimização. A escolha pela utilização da versão estendida ao invés do Algoritmo 4 exige o conhecimento a priori da presença de oclusões nas imagens de teste. É de se esperar que a utilização da versão estendida nos casos onde não há oclusões apresente resultados similares caso fosse usado o Algoritmo 4. Entretanto o custo computacional associado é maior. Portanto, se não há conhecimento da presença de oclusões e o custo computacional não é um problema, sugere-se sempre utilizar a versão estendida. Por outro lado, se o custo computacional for uma restrição, ao utilizar o Algoritmo 4 deve-se considerar que para os casos de oclusões o desempenho pode não ser satisfatório.

Embora a componente esparsa da decomposição JSM-MC carregue de fato as informações esparsa das imagens, observou-se, nesta dissertação, que tais informações não são peculiares na definição da classe. Em geral, as componentes esparsas carregam ruídos esparsos que por ventura a imagem tenha. Logo, o sistema proposto apresenta uma vantagem quando as imagens de treino possuem oclusão uma vez que a matriz

esparsa do JSM-MC não é utilizada. De certa forma, a não utilização da matriz esparsa imprime a função de um filtro, principalmente quando as imagens de treino possuem ruídos esparsos, já que estes são bem representados pela matriz esparsa. Outro ponto que vale a pena ser ressaltado é que não há necessidade de uma escolha diferenciada na composição da base de treino, como a abordagem feita por Wright et al. [54]. Apesar do JSM-MC isolar parte das variações de iluminação em suas matrizes de posto pequeno, a utilização da equalização de histograma mostrou-se como etapa importante para melhora na taxa de acerto na classificação.

A modificação no critério de classificação pela utilização da norma- $\ell_1$  mostrou-se coerente com as observações feitas da não utilização das matrizes esparsas, já que a utilização das matrizes esparsa para compor a base de treino poderia deixar a solução da representação menos esparsa (por serem esparsas, poderiam contribuir em qualquer classe, resultando em um vetor de representação menos esparso). Ademais, os resultados apresentados mostram que a abordagem escolhida obteve uma maior taxa de acerto na classificação quando comparada ao critério de classificação utilizando os resíduos.

O agrupamento, apesar de não ser uma etapa fundamental para o funcionamento do sistema, possibilitou um menor tempo de execução dos algoritmos de otimização. Ademais, em testes realizados, verificou-se que pouca diferença é observada na taxa de acerto ao usar todas as matrizes de posto pequeno nos casos onde não há oclusão. Já nos casos de oclusão, observou-se que um número maior de agrupamento possibilitou maiores taxas de acerto na classificação. Um estudo mais detalhado do impacto do agrupamento na taxa de classificação fica como sugestão para trabalhos futuros.

Os resultados apresentados nos testes de rejeição de imagens de teste tiveram resultados satisfatórios. Vale ressaltar, que as imagens de teste utilizadas como não face foram faces humanas (com e sem oclusões e que não pertenciam à nenhuma classe do conjunto de treino) e que, de certa forma, apresentam similaridades com as imagens de treino, já que são faces humanas. Portanto, de certa forma, este seria uma das piores situações para o sistema de rejeição, já que qualquer outra imagem de teste, que não seja face humana, apresentaria menor semelhança com as imagens de treino e portanto os coeficientes da representação esparsa estariam mais espalhados e maior seriam as chances dessa imagem ser rejeitada. Contudo, isso é uma característica de sistemas baseados em representação esparsa para classificação conforme já observado por Wright et al. [54]. É possível observar, nos testes realizados, que a escolha do parâmetro  $\tau$ , seja qual for a aplicação, não é uma tarefa trivial, já que este parâmetro está fortemente ligado ao conjunto de dados utilizado. Uma metodologia para determinação deste parâmetro de forma automática para diferentes aplicações, pode ser um tema para um estudo posterior.

Ainda como possibilidades de melhoria no sistema, sugere-se a implementação dos algoritmos em C, possibilitando assim, um menor tempo de execução no reconhecimento de faces. Como parte do objetivo inicial dessa dissertação é comparar o sistema proposto com o algoritmo apresentado por Wright et al. [54], buscou-se utilizar as mesmas base de dados. Contudo, não foi possível utilizar a base de dados AR uma vez que, até o momento da apresentação da dissertação, não houve disponibilidade, por parte dos criadores da base de dados, para liberação dos dados. Portanto, para projetos futuros, sugere-se a aplicação dos algoritmos nessa base dados para posterior comparação com os resultados apresentados por [54], conforme proposta inicial da dissertação.

Embora se tenha tratado algumas das variabilidades encontradas no problema de reconhecimento de faces, uma abordagem diferenciada ainda se faz necessária para tratar o caso de variação de escala e rotação. Portanto, outra sugestão para trabalhos futuros refere-se à incorporação de uma abordagem invariante a escala e rotação ao método proposto.

# Referências Bibliográficas

- [1] AMALDI, E. E KANN, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237 – 260.
- [2] BARON, D., DUARTE, M. F., WAKIN, M. B., SARVOTHAM, S., E BARANIUK, R. G. (2009). Distributed compressive sensing. *CoRR*, abs/0901.3403.
- [3] BELHUMEUR, P. N., HESPANHA, J. P., E KRIEGMAN, D. J. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection. 19(7):711–720.
- [4] BERTSEKAS, D. (1995). *Nonlinear programming*. Optimization and neural computation series. Athena Scientific.
- [5] BERTSEKAS, D. (1996). *Constrained optimization and Lagrange multiplier methods*. Optimization and neural computation series. Athena Scientific.
- [6] BISHOP, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [7] CAI, J.-F., CANDÈS, E. J., E SHEN, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.
- [8] CANDÈS, E. (2006). Compressive sampling. Em *Proceedings of the International Congress of Mathematicians, Madrid, Spain*, volume 3, páginas 1433–1452.
- [9] CANDÈS, E., LI, X., MA, Y., E WRIGHT, J. (2010a). Robust principal component analysis? *Journal of the ACM*.
- [10] CANDÈS, E., LI, X., MA, Y., E WRIGHT, J. (2010b). Robust principal component analysis? Recovering low-rank matrices from sparse errors. Em *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, páginas 201–204.
- [11] CANDÈS, E. E RECHT, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772.

- [12] CANDÈS, E. J., ROMBERG, J., E TAO, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509.
- [13] CANDÈS, E. J. E TAO, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425.
- [14] CHEN, C.-T. (1999). *Linear System Theory and Design*. Oxford University Press, 3a edição.
- [15] CHEN, S. E DONOHO, D. (1994). Basis pursuit. Em *Proc. Conf Signals, Systems and Computers Record of the Twenty-Eighth Asilomar Conf*, volume 1, páginas 41–44.
- [16] CHEN, S. S., DONOHO, D. L., MICHAEL, E SAUNDERS, A. (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61.
- [17] COTTER, S. F. (2010). Sparse representation for accurate classification of corrupted and occluded facial expressions. Em *Proc. IEEE Int Acoustics Speech and Signal Processing (ICASSP) Conf*, páginas 838–841.
- [18] DONOHO, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.
- [19] DU, S. E WARD, R. (2006). Adaptive region-based image enhancement method for face recognition under varying illumination conditions. Em *Proc. IEEE Int Acoustics, Speech and Signal Processing Conf. ICASSP 2006*, volume 2.
- [20] DUDA, R. O., HART, P. E., E STORK, D. G. (2001). *Pattern Classification*. Wiley-Interscience, 2 edição.
- [21] ETEMAD, K. E CHELLAPPA, R. (1996). Face recognition using discriminant eigenvectors. Em *Proc. Conf. IEEE Int Acoustics, Speech, and Signal Processing ICASSP-96*, volume 4, páginas 2148–2151.
- [22] FORNASIER, M. E RAUHUT, H. (2011). Compressive sensing. Em Scherzer, O., editor, *Handbook of Mathematical Methods in Imaging*, páginas 187–228. Springer.
- [23] FRANK, A. E ASUNCION, A. (2010). UCI machine learning repository.
- [24] GANESH, A., WAGNER, A., WRIGHT, J., YANG, A., ZHOU, Z., E MA, Y. (2011). *Face recognition by sparse representation. Compressed Sensing: Theory and Applications*. Cambridge University Press.

- [25] GEORGHIADES, A. S., BELHUMEUR, P. N., E 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.
- [26] GONZÁLEZ, R. E WOODS, R. (2002). *Digital image processing*. Prentice Hall.
- [27] GRANT, M. E BOYD, S. (2008). Graph implementations for nonsmooth convex programs. Em Blondel, V., Boyd, S., e Kimura, H., editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, páginas 95–110. Springer-Verlag Limited. [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).
- [28] GRANT, M. E BOYD, S. (2011). CVX: Matlab software for disciplined convex programming, version 1.21. <http://cvxr.com/cvx>.
- [29] HE, X., YAN, S., HU, Y., NIYOGI, P., E ZHANG, H.-J. (2005). Face recognition using laplacianfaces. 27(3):328–340.
- [30] HSIEH, P.-C. E TUNG, P.-C. (2009). Illumination-robust face recognition using an efficient mirror technique. Em *Proc. 2nd Int. Congress Image and Signal Processing CISP '09*, páginas 1–5.
- [31] KANADE, T. (1973). *Picture Processing System by Computer Complex and Recognition of Human Faces*. Tese de Doutorado, Kyoto University.
- [32] LADES, M., VORBRUGGEN, J. C., BUHMANN, J., LANGE, J., VON DER MALS-BURG, C., WURTZ, R. P., E KONEN, W. (1993). Distortion invariant object recognition in the dynamic link architecture. 42(3):300–311.
- [33] LEE, K.-C., HO, J., E KRIEGMAN, D. J. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698.
- [34] LI, S. E JAIN, A. (2011). *Handbook of Face Recognition*. Springer, 2 edição.
- [35] LIN, Z., CHEN, M., E WU, L. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Mathematical Programming*, math.OC.
- [36] LIU, C.-C. E CHIN, S.-Y. (2010). A face identification algorithm using support vector machine based on binary two dimensional principal component analysis. Em *Proc. IET Int Frontier Computing. Theory, Technologies and Applications Conf*, páginas 193–198.



- [37] LUSTIG, M., DONOHO, D., E PAULY, J. M. (2007). Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58:1182–1195.
- [38] LYONS, M., AKAMATSU, S., KAMACHI, M., E GYOBA, J. (1998). Coding facial expressions with gabor wavelets. Em *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, páginas 200 –205.
- [39] MALLAT, S. (2009). *A Wavelet Tour of Signal Processing: The Sparse Way*. Elsevier Inc., 3a edição.
- [40] MALLAT, S. G. E ZHANG, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415.
- [41] MARTINEZ, J. E SANTOS, S. (1995). *Métodos Computacionais De Otimização*. Colóquio Brasileiro de Matemática - Apostilas. IMPA.
- [42] MENDONÇA, M. M. (2008). Comparação entre métodos de normalização da iluminação utilizados para melhorar a taxa do reconhecimento facial. Dissertação de Mestrado, Universidade de São Paulo - Escola de Engenharia de São Carlos - Departamento de Engenharia Elétrica.
- [43] MENDONÇA, M. M., DENIPOTE, J. G., FERNANDES, R. A. S., E PAIVA, M. S. V. D. (2007). Illumination normalization methods for face recognition. Porto Alegre. Brazilian Symposium on Computer Graphics and Image Processing, 20. (SIB-GRAPI), Sociedade Brasileira de Computação.
- [44] PATEL, V. M., CHELLAPPA, R., E TISTARELLI, M. (2010). Sparse representations and random projections for robust and cancelable biometrics. Em *Proc. 11th Int Control Automation Robotics & Vision (ICARCV) Conf*, páginas 1–6.
- [45] PILLAI, J., PATEL, V., CHELLAPPA, R., E RATHA, N. (2011). Secure and robust iris recognition using random projections and sparse representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99).
- [46] RABBANI, M. E JONES, P. W. (1991). *Digital Image Compression Techniques*, volume TT07 of *Tutorial texts in optical engineering*. Society of Photo-Optical Instrumentation Engineers (SPIE) Press Book, Bellingham, WA, USA, 1st edição.
- [47] SAMMON, J. W., J. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, (5):401–409.

- [48] SELLAHEWA, H. E JASSIM, S. A. (2010). Image-quality-based adaptive face recognition. *IEEE Transactions on Instrumentation and Measurement*, 59(4):805–813.
- [49] SIROVICH, L. E KIRBY, M. (1987). Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524.
- [50] STARCK, J.-L., MURTAGH, F., E FADILI, J. M. (2010). *Sparse Image and Signal Processing: Wavelet, Curvelet, Morphological Diversity*. Cambridge University Press.
- [51] TURK, M. E PENTLAND, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.
- [52] VIOLA, P. E JONES, M. (2001). Rapid object detection using a boosted cascade of simple features. Em *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2001*, volume 1.
- [53] WISKOTT, L., FELLOUS, J.-M., KRUGER, N., E VON DER MALSBERG, C. (1997). Face recognition by elastic bunch graph matching. Em *Proc. Conf. Int Image Processing*, volume 1, páginas 129–132.
- [54] WRIGHT, J., YANG, A. Y., GANESH, A., SASTRY, S. S., E MA, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227.
- [55] XU, M., ZHENG, C., ZHANG, L., E VINCENT, N. T.-Y. (2009). Tumor classification via sparse representation based on metasample. Em *Proc. Second Int. Symp. Knowledge Acquisition and Modeling KAM '09*, volume 1, páginas 31–34.
- [56] YANG, A. Y., WRIGHT, J., MEMBER, S., MA, Y., MEMBER, S., E SASTRY, S. S. (2007). Feature selection in face recognition: A sparse representation perspective. Relatório Técnico UCB/EECS-2007-99, EECS Department, University of California, Berkeley.
- [57] YUAN, X. E YANG, J. (2009). Sparse and low-rank matrix decomposition via alternating direction methods. *Convergence*, páginas 1–11.
- [58] ZHANG, Q. E LI, B. (2010). Joint sparsity model with matrix completion for an ensemble of face images. Em *Proc. 17th IEEE Int Image Processing (ICIP) Conf*, páginas 1665–1668.
- [59] ZHAO, W., CHELLAPPA, R., PHILLIPS, P. J., E ROSENFELD, A. (2003). Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458.

- 
- [60] ZHENG, C., ZHANG, L., NG, T., E SHIU, C. (2011). Metasample based sparse representation for tumor classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, (99).

# Anexo A

## Equalização de Histograma

O histograma de uma imagem digital com níveis de cinza no intervalo  $[0, L - 1]$  é uma função discreta  $h(r_k) = n_k$ , onde  $r_k$  é o  $k$ -ésimo nível de cinza e  $n_k$  é o número de *pixels* com nível de cinza  $r_k$  que a imagem possui. É uma prática comum normalizar o histograma, dividindo cada um de seus valores pelo número total de *pixels*,  $n$ , na imagem. Logo, a normalização do histograma é dado por  $p(r_k) = n_k/n$ , para  $k = 0, \dots, L - 1$ , que em geral é considerada como uma estimativa da probabilidade de ocorrência de níveis de cinza  $r_k$  [26].

A equalização de histograma é uma técnica de processamento de imagens que é usada para melhorar o contraste de imagens. Consiste em redistribuir os níveis de cinza em uma imagem de tal forma que todos os níveis de cinza dentro da imagem ocorram quase igualmente. Portanto, deve-se ter uma função de transformação,  $T(r)$ , que mapeia a função de densidade de probabilidade do histograma de uma imagem em uma função de densidade de probabilidade uniformemente distribuída.

Uma das formas, comumente utilizada, para realizar a equalização de histograma é utilizar a função de distribuição acumulada (CDF) da distribuição de probabilidade dos níveis de cinza da imagem a ser realçada dada por [26]

$$s_k = T(r_k) = \sum_{j=0}^k \left( \frac{n_j}{n} \right) = \sum_{j=0}^k p(r_j) \quad (\text{A.1})$$

onde  $k = 0, 1, \dots, L - 1$ .

# Anexo B

## *K-means clustering*

Considere um conjunto de dados  $\chi = \{\mathbf{x}_1, \dots, \mathbf{x}_n, \dots, \mathbf{x}_N\}$  com  $N$  observações da variável aleatória  $\mathbf{x}$  e que deseja-se identificar grupos, ou *clusters*, a partir desses dados. Portanto, o objetivo é aglomerar os dados em um número  $K$  de *clusters*, com  $K$  conhecido. Suponha agora vetores  $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k, \dots, \boldsymbol{\mu}_K\}$ , da mesma dimensão de  $\mathbf{x}$ , sendo o centro dos *clusters*. Então, deve-se achar os elementos de  $\chi$  que possuem menor distâncias entre o centro de um *cluster*,  $\boldsymbol{\mu}_k$ , comparadas com as distâncias aos demais centros. O objetivo do algoritmo é então atribuir os elementos de  $\chi$  aos *clusters*, bem como determinar um conjunto de vetores  $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k, \dots, \boldsymbol{\mu}_K\}$ , tal que a soma dos quadrados das distâncias de cada elemento de  $\chi$  ao vetor mais próximo  $\boldsymbol{\mu}_k$  é mínimo.

Considere, para cada ponto  $\mathbf{x}_n$ , um indicador binário de variáveis  $r_{nk} \in \{0, 1\}$ , onde  $k = 1, \dots, K$  descrevendo quais dos  $K$  *clusters* o ponto  $\mathbf{x}_n$  é atribuído, de tal forma que se o elemento  $\mathbf{x}_n$  é atribuído ao *cluster*  $k$  então  $r_{nk} = 1$  e  $r_{nj} = 0$  para  $j \neq k$ . Isso é conhecido como esquema de codificação 1-de- $K$ . Pode-se então definir a função objetivo, algumas vezes chamada de medida de distorção, dada por [6]

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \quad (\text{B.1})$$

que representa a soma dos quadrados das distâncias de cada ponto ao seu respectivo vetor atribuído  $\boldsymbol{\mu}_k$ . O objetivo é então encontrar valores de  $\{r_{nk}\}$  e  $\{\boldsymbol{\mu}_k\}$  de tal forma que  $J$  seja mínimo.

Isto pode ser feito através de um procedimento iterativo em duas etapas sucessivas. Primeiro escolhe-se valores iniciais para  $\boldsymbol{\mu}_k$ . Então na primeira etapa  $J$  é minimizado com respeito a  $r_{nk}$ , deixando  $\boldsymbol{\mu}_k$  fixo. Na segunda etapa,  $J$  é minimizado com respeito a  $\boldsymbol{\mu}_k$  deixando  $r_{nk}$  fixo. Este procedimento é repetido até obter convergência.

## Anexo C

# Decomposição em Valores Singulares (SVD)

Toda matriz  $L$ , dimensão  $m \times n$  e posto  $r$ , pode ser transformada na seguinte forma [14]:

$$L = U\Sigma V' \tag{C.1}$$

com  $U$  e  $V$  sendo matrizes de dimensão  $m \times m$  e  $n \times n$  respectivamente,  $U'U = UU' = I_m$ ,  $V'V = VV' = I_n$  e  $\Sigma$  uma matriz  $m \times n$  com os valores singulares na diagonal.

As colunas de  $V$  são os autovetores ortonormalizados de  $L'L$  e as colunas de  $U$  são os autovetores ortonormalizados de  $LL'$ . Uma vez que  $U$ ,  $\Sigma$  e  $V$  são determinados, o posto de  $L$  é igual ao número de valores singulares não nulos. As primeiras  $r$  colunas de  $U$  são uma base ortonormal da imagem (espaço range) de  $L$ . As últimas  $(n - r)$  colunas de  $V$  são uma base ortonormal do espaço nulo de  $L$ .

# Anexo D

## *Robust PCA*

Suponha uma matriz  $M$  e que deseja-se uma decomposição conforme

$$M = L_0 + S_0, \quad (\text{D.1})$$

onde  $L_0$  possui posto pequeno e  $S_0$  é esparsa. Em PCA Robusto [10] este problema pode ser resolvido pelo problema de otimização:

$$\begin{aligned} \min \quad & \|L\|_* + \lambda \|S\|_1 \\ \text{sujeito a} \quad & L + S = M \end{aligned} \quad (\text{D.2})$$

com  $\lambda$  sendo uma constante. Para que o problema de otimização (D.2) retorne a resposta desejada, duas condições devem ser atendidas: primeiro, a matriz de posto pequeno,  $L_0$ , não deve ser esparsa e segundo, a matriz esparsa,  $S_0$ , não pode ter posto pequeno. Para a solução de (D.2), Candès et al. [10] sugerem a utilização do lagrangiano aumentado. O método lagrangiano aumentado opera sobre a função lagrangiana aumentada:

$$\mathcal{L}(L, S, \Lambda) = \|L\|_* + \lambda \|S\|_1 + \langle \Lambda, M - L - S \rangle + \frac{\mu}{2} \|M - L - S\|_F^2 \quad (\text{D.3})$$

Este problema pode ser resolvido por algoritmos de multiplicadores de Lagrange fazendo, repetidamente,  $(L_k, S_k) = \arg \min_{L, S} \mathcal{L}(L, S, \Lambda_k)$ , e então atualizar a matriz dos multiplicadores de Lagrange usando  $\Lambda_{k+1} = \Lambda_k + \mu(M - L_k - S_k)$ . Entretanto, é possível evitar a solução de uma sequência de programas convexos reconhecendo que  $\min_L \mathcal{L}(L, S, \Lambda)$  e  $\min_S \mathcal{L}(L, S, \Lambda)$  possuem soluções simples e eficientes.

Seja  $S_\tau(x)$  o operador *soft thresholding* conforme definido em e  $\mathcal{D}_\tau(L)$  o operador *Singular Value Trhesholding* conforme (2.17), então

$$\arg \min_S \mathcal{L}(L, S, \Lambda) = \mathcal{S}_{\lambda\mu^{-1}}(M - L - \mu^{-1}\Lambda) \quad (\text{D.4})$$

e

$$\arg \min_L \mathcal{L}(L, S, \Lambda) = \mathcal{D}_{\mu^{-1}}(M - L - \mu^{-1}\Lambda). \quad (\text{D.5})$$

Então, uma estratégia para resolver (D.2) é primeiro minimizar  $\mathcal{L}$  com respeito à  $L$  (fixando  $S$ ), depois minimizar  $\mathcal{L}$  como respeito à  $S$  (fixando  $L$ ) e então atualizar a matriz dos multiplicadores de Lagrange,  $\Lambda$ , baseado no resíduo  $M - L - S$ . Esta estratégia é indicada no Algoritmo 6.

---

**Algoritmo 6:** PCA Robusto [10, 35, 57]

---

**Inicialização:**  $S_0 = Y_0 = 0$ ,  $\mu > 0$

**enquanto** *não convergir* **faça**

$$\left| \begin{array}{l} L_{k+1} = \mathcal{D}_{\mu^{-1}}(M - S_k + \mu^{-1}\Lambda_k); \\ S_{k+1} = \mathcal{S}_{\lambda\mu^{-1}}(M - L_{k+1} + \mu^{-1}\Lambda_k); \\ Y_{k+1} = Y_k + \mu(M - L_{k+1} - S_{k+1}); \end{array} \right.$$

**fim**

**Saída:**  $L, S$

---