UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FELIPE JOSÉ COELHO PEDROSO

RECONHECIMENTO AUTOMÁTICO DE EXPRESSÕES FACIAIS BASEADO EM MODELAGEM ESTATÍSTICA

VITÓRIA
2013

FELIPE JOSÉ COELHO PEDROSO

RECONHECIMENTO AUTOMÁTICO DE EXPRESSÕES FACIAIS BASEADO EM MODELAGEM ESTATÍSTICA

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: Evandro Ottoni Teatini Salles.

VITÓRIA
2013

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

Pedroso, Felipe José Coelho, 1985-

P372r Reconhecimento automático de expressões faciais baseado em modelagem estatística / Felipe José Coelho Pedroso. – 2013.

115 f. : il.

Orientador: Evandro Ottoni Teatini Salles.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Processamento de imagens. 2. Interação homem-máquina. 3. Sistemas de reconhecimento de padrões. 4. Expressão facial. 5. Aprendizado do computador. I. Salles, Evandro Ottoni Teatini. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 621.3

FELIPE JOSÉ COELHO PEDROSO

RECONHECIMENTO AUTOMÁTICO DE EXPRESSÕES FACIAIS BASEADO EM MODELAGEM ESTATÍSTICA

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 28 de março de 2013.

COMISSÃO EXAMINADORA

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

Profa. Dra. Raquel Frizera Vassallo
Universidade Federal do Espírito Santo

Prof. Dr. João Marques Salomão
Instituto Federal do Espírito Santo

Aos meus pais.

To give anything less than your best is to sacrifice the gift.

Steve Prefontaine

Agradecimentos

Aos meu pais, Camilo e Eliane e meu irmão Daniel pelo apoio e incentivo incondicional.

Ao orientador Evandro. Orientador, incentivador e exemplo.

Aos familiares, avós, tios, primos que, não importando a distância, sempre se manifestaram como torcida genuína pelo sucesso.

Aos amigos que sempre apoiaram, incluindo os de outras áreas de conhecimento.

Aos colegas de laboratório e da pós-graduação.

Aos colegas de trabalho.

A todos que colaboraram direta ou indiretamente na realização dessa pesquisa, pois cada pessoa e evento moldaram os caminhos que levaram ao resultado final dessa dissertação.

Aos professores que me acompanharam durante a graduação e aos professores do programa de pós-graduação.

A CAPES e ao PPGEE - UFES pelo incentivo, suporte e financiamento à pesquisa.

Aos que colaboraram com imagens e banco de dados utilizados nos testes.

Sumário

Lista de Tabelas

Lista de Figuras

Resumo

As expressões faciais são alvos constante de estudos desde Charles Darwin, em 1872. Pesquisas na área de psicologia e, em destaque, os trabalhos de Paul Ekman afirmam que existem expressões faciais universais básicas e elas são manifestadas em todos os seres humanos independente de fatores como gênero, idade, cultura e ambiente social. Ainda pode-se criar novas expressões mais complexas combinando as expressões fundamentais de alegria, tristeza, medo, nojo, raiva, surpresa e desprezo, além da face neutra. O assunto ainda é atual, uma vez que há uma grande necessidade de implementar interfaces homem-máquinas (IHM) capazes de identificar a expressão de um indivíduo e atribuir uma saída condizente com a situação observada. Pode-se citar como exemplos iterações homem-robô, sistemas de vigilância e animações gráficas. Nesse trabalho é proposto um sistema automático para identificar expressões faciais. O sistema é dividido em três etapas: localização de face, extração de características e identificação da expressão facial. O banco de dados *Japanese Facial Expression Database* - JAFFE foi utilizado para treinamentos e testes . A localização da face é realizada de maneira automática através do *framework* proposto por Viola-Jones e é estimado o centro da face. Na sequência, utiliza-se o algoritmo *Active Appearance Model* - AAM para descrever estatisticamente um modelo de forma e textura para o banco de dados. Com esse descritor é possível gerar um vetor de aparência capaz de representar, com redução de dimensão, uma face e, consequentemente, a expressão facial contida nela através de um algoritmo iterativo de busca a partir de um modelo médio. Esse vetor é utilizado na etapa de reconhecimento das expressões faciais, onde são testados os classificadores baseados no vizinho mais próximo k-NN e a máquina de vetores de suporte - SVM com *kernel* RBF para tratar o problema de forma não linear. É proposto um mecanismo de busca na saída do bloco de detecção de faces para diminuir o erro do modelo, pois o sucesso do algoritmo é altamente dependente do ponto inicial de busca. Também é proposto uma mudança no algoritmo AAM para redução do erro de convergência entre a imagem real e o modelo sintético que a representa, abordando o problema de forma não linear. Testes foram realizados utilizando a validação cruzada *leave one out* para todas as expressões faciais e o classificador SVM-RBF. O sistema apresentou um taxa de acerto de 55,4%, com sensibilidade 60,25% e especificidade 93,95%.

Abstract

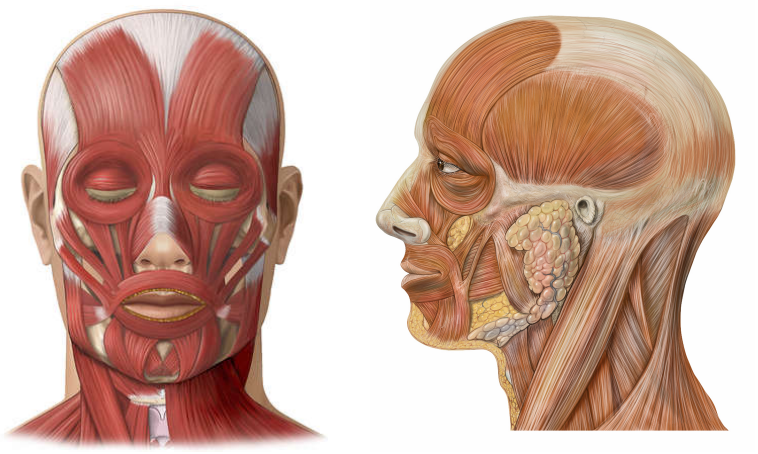
Facial expressions are constant targets of studies since Charles Darwin in 1872. Research in psychology and highlighted the work of Paul Ekman claim that there are universal basic facial expressions and they are expressed in all human beings regardless of factors such as gender, age, culture and social environment. Although you can create new more complex expressions combining the fundamental expressions of happiness, sadness, fear, disgust, anger, surprise and contempt, beyond the neutral face. The matter is still relevant, since there is a great need to implement human machine interfaces (HMI) able to identify the expression of an individual and assign an output consistent with the observed situation. One can cite as examples iterations man-robot surveillance and motion graphics. In this work it's proposed an automatic system to identify facial expressions. The system is divided into three blocks: face localization, feature extraction and identification of facial expression. The Japanese Facial Expression Database - JAFFE was used for training and testing. The location of the face is done automatically using the framework proposed by Viola and Jones estimating center of the face. Following the Active Appearance Model - AAM algorithm is used to describe statistical model of shape and texture to the database. With this descriptor is possible to generate a vector capable of representing faces with reduced dimension and hence the facial expression contained therein through an iterative search algorithm from an average model. This vector is used in recognizing facial expressions block, where the classifiers are tested based on the nearest neighbor k-NN and support vector machine - SVM with RBF kernel to address the problem of non-linear way. A mechanism to decrease the error of the model is proposed before the output of the face detection block, because the success of the algorithm is highly dependent on the starting point of the search. A change in the AAM algorithm is also proposed to reduce the convergence error between actual and synthetic model that is addressing the problem of nonlinear way. Tests were conducted using leave one out cross validation for all the facial expressions and the final classifier was SVM-RBF. The system has an accuracy rate of 55.4%, with 60,25% sensitivity and 93,95% specificity.

Capítulo 1

Introdução

1.1 Motivação

Os seres humanos podem expressar suas ideias, sentimentos e reações através de diferentes formas de comunicação. A expressão facial é uma forma de comunicação não verbal resultante de determinadas configurações ou contrações dos músculos faciais que provocam modificações e deformações na face (FASEL; LUETTIN, 2003).A face possui 53 músculos divididos em 5 grandes grupos: região da testa, ao redor dos olhos, bochecha, abaixo e acima dos olhos, conforme ilustra a Figura 1.1.

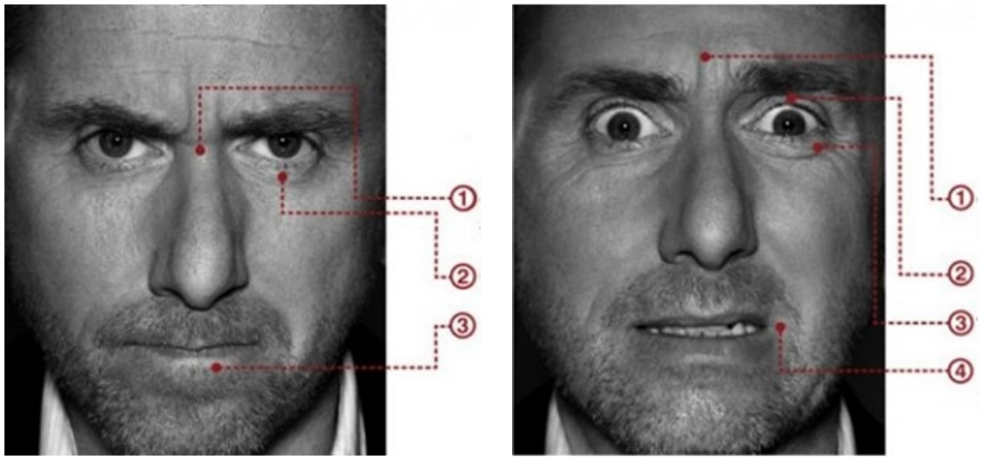


(a) (b)

Figura 1.1: Músculos que compõem a face em vista (a) frontal e (b) lateral.

Essas modificações podem ser no formato de elementos que compõem a face como, por

exemplo, a boca aberta ou fechada e/ou modificações na textura como a presença de protuberâncias devido a contrações na testa, conforme Figura 1.2. As informações contidas nas expressões faciais são altamente representativas. Em um processo de conversação as palavras contribuem com 7% da transmissão de informação entre ouvinte e interlocutor. A entonação de voz representa 38% da informação e a expressão facial corresponde a 55% (MEHRABIAN, 1968). Em situções comuns do cotidiano é visível a discrepância de informação entre a comuniação verbal e as expressões faciais, indicando a presença de uma anormalidade.

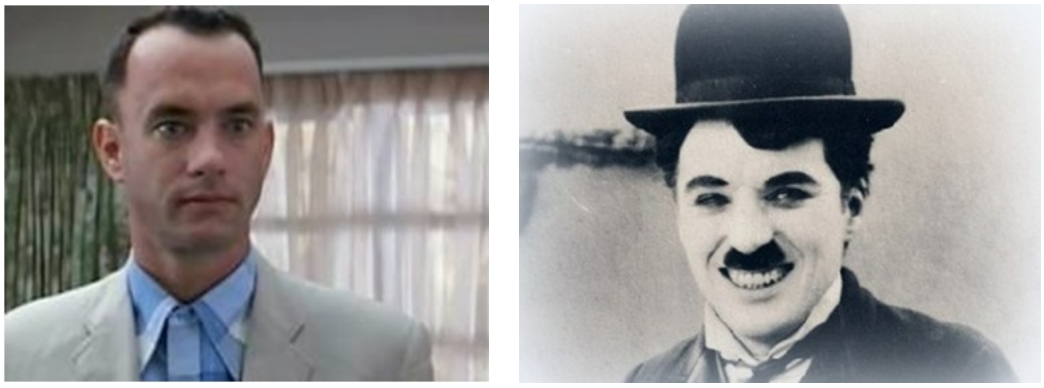


(a) (b)

Figura 1.2: Expressões faciais formadas através de deformações no formato ou na textura de elementos que compõem a face. (a)Rugas no nariz, olhos e contração da boca. (b)Rugas nos olhos e testa , abertura excessiva dos olhos e assimetria na boca.

Diferentes emoções são caracterizadas por diferentes expressões faciais e, consequentemente, diferentes movimentos e deformações da face. O sorriso e as sobrancelhas levantadas contêm informações sobre a emoção do indivíduo que a expressa. Um indivíduo que demonstra surpresa apresenta uma abertura de boca diferente de um indivíduo feliz. Já uma pessoa com raiva apresenta rugosidades na testa diferentes de uma pessoa que demonstra o desprezo. Em 1872 Darwin (DARWIN, 1872) já realizava pesquisas sobre a universalidade das expressões faciais e estudos posteriores como o de Paul Ekman (EKMAN; FRIESEN, 1971) estabeleceram uma corrente de que as expressões faciais são universais. A Figura 1.3 demonstra como os seres humanos possuem a capacidade de identificar as expressões faciais independentemente de fatores como a cultura, idade e gênero. Ou seja, todos os indivíduos se expressam com as mesmas características faciais. No entanto, situações iguais podem gerar expressões faciais diferentes em função do indivíduo de teste e diferentes interpretações do ambiente, dependendo do codificador e decodificador, ou seja, a pessoa que executa a expressão facial e a pessoa que a observa (MATSUMOTO, 1993). Por exemplo, um ato que gera felicidade em indivíduos de uma cultura pode gerar tristeza em indivíduos de um outro

grupo. Apesar disso, os dois grupos apresentam de forma similar as expressões de felicidade e de tristeza.



(a) (b)

Figura 1.3: Expressões faciais possuem universais, possibilitando sua utilização em Interfaces Homem Máquina - IHM. (a) Surpresa. (b) Felicidade.

Em (EKMAN, 1994) foram propostas seis emoções básicas: alegria, raiva, medo, tristeza, surpresa e nojo/aversão além da expressão neutra, caracterizada pela ausência de expressão facial. A combinação das expressões básicas é capaz de gerar emoções mais complexas. Estudos posteriores ainda incluem o desprezo como uma sétima expressão básica universal (EKMAN; HEIDER, 1988).

O problema de reconhecer uma expressão facial é dual ao reconhecer uma face, pois no primeiro é desejado reconhecer a expressão independentemente do indivíduo enquanto no segundo a identidade do indivíduo é o alvo independente da sua expressão facial (CHIBELUSHI; BOUREL, 2003).

O aumento do processamento computacional e as novas necessidades da sociedade impulsionam o desenvolvimento e aperfeiçoamento de sistemas inteligentes (CALABRESE et al., 2011), incluindo o campo da computação pervasiva[1](SATYANARAYANAN, 2001). Interfaces homem-máquina (IHM) são indispensáveis e, para um correto funcionamento, muitas vezes é necessário que a máquina compreenda as expressões ou emoções de seus utilizadores para atender às suas demandas e expectativas. Portanto, um sistema que realize o reconhecimento de expressões faciais pode ser utilizado nas mais diversas áreas incluindo:

– Interação com um robô: facilitar o entendimento das tarefas e humanizar a máquina, como na Figura 1.4(a);

[1]A computação pervasiva ou computação ubíqua relaciona a utilização de técnicas computacionais na sociedade humana com objetivo de tornar as iterações homem-máquina elementos inerentes às atividades da sociedade, tornando o sistema de informática onipresente.

– Monitoramento de Atividades: verificar desempenho do utilizador em determinadas tarefas como no ensino à distância (LONGHI; BERCHAT; BEHAR, 2007) (vide Figura 1.4(b));

– Segurança: detectar reações suspeitas como, por exemplo, a raiva em locais estratégicos ou em aglomerações utilizando câmeras, conforme Figura 1.4(c);

– Campanhas de *marketing*: verificar reação dos clientes ao utilizar ou ver um produto;

– Animação Gráfica: transpor as emoções corretas para um personagem virtual;

– Estudos em Medicina e Psicologia: suporte ao monitoramento de pacientes e em pesquisas.

(a) (b) (c)

Figura 1.4: A universalidade das expressões faciais possibilitam sua utilização em Interfaces Homem Máquina. (a) Interação com um robô. (b) Monitoramento de ativiadades. (c)Segurança.

É importante ressaltar que a identificação de expressões faciais é diferente de identificação de emoções, uma vez que a segunda depende da primeira e de outros fatores como entonação da voz, gestos e pose.

O reconhecimento da expressão facial pode considerar três fatores. Primeiramente, o local de ocorrência da deformação na forma e/ou textura dos elementos da face como os olhos e boca. Em seguida, a duração da deformidade como a persistência de boca aberta ou pequenos movimentos de franzir o canto da boca. E, por fim, a intensidade da deformação como grandes elevações de sobrancelha ou formação de pequenas rugas na testa. Portanto, trabalhar com imagens estáticas gera uma maior dificuldade na classificação. Esse problema é normalmente contornado utilizando bancos de imagens com expressões em poses forçadas.

1.2 Objetivos

Desenvolver um sistema automático de reconhecimento de expressões faciais que deve ser capaz de localizar uma face para posterior determinação da expressão facial em imagens estáticas.

O trabalho proposto é dividido em três etapas: detecção de face, extração de características e classificação das expressões faciais, conforme Figura 1.5.

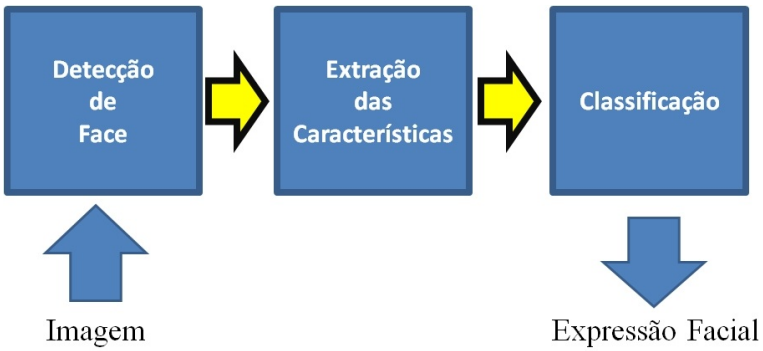


Figura 1.5: Diagrama de blocos do trabalho.

O processo de detecção de faces é realizado utilizando um modelo estatístico que permite realizar outras funções além da classificação de expressões. Somam-se aos objetivos do trabalho a geração de imagens sintéticas que representem os indivíduos de um banco de dados incluindo poses não conhecidas e a localização de olhos, nariz e boca, que podem ser utilizados em outras aplicações que demandam a localização precisa de tais pontos.

1.3 Caracterização do Problema

Uma IHM eficiente na tarefa de identificação de expressões faciais deve ser capaz de localizar a face com exatidão independentemente da complexidade de cenário. Portanto, problemas como iluminação, escala, rotação, translação, oclusão, múltiplos alvos e qualidade da imagem de aquisição devem ser contornados. Nesse sentido, o *framework* Viola-Jonas, proposto em (VIOLA; JONES, 2001), mostra-se robusto para essa tarefa.

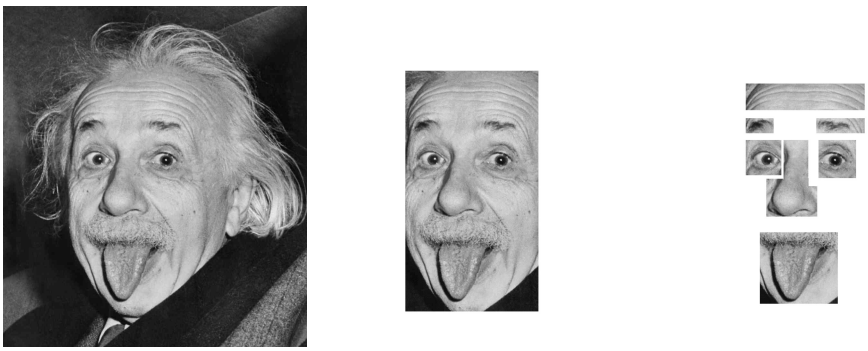
Também observa-se a necessidade de redução de dimensionalidade, uma vez que a informação contida em uma imagem torna-se excessiva para um classificador. Uma imagem de 512×512 *pixels* de resolução, se utilizada integralmente, gera um vetor de dimensão

1×262.144. Essa observação vai ao encontro da necessidade de redução de custos computacionais para a implementação em tempo real do sistema e para um menor esforço do classificador. A análise de componentes principais - PCA inserida no AAM - *Active Appearance Model*é uma forma de minimizar o problema.

Como existem múltiplas expressões faciais, o classificador deve ser capaz de resolver um problema multi-classe. O classificador k-NN (*Nearest Neighbors*) trabalha com estimativas de densidade de probabilidade de cada classe ao passo que o Support Vector Machine - SVM trabalha dividindo o problema no formato uma classe x todas as outras.

1.4 Estado da Arte

O reconhecimento de expressões faciais está em constante desenvolvimento e ainda é um tema atual. Diversas técnicas foram desenvolvidas ao longo dos anos e diferentes abordagens realizadas (PANTIC; ROTHKRANTZ, 2000), (CHIBELUSHI; BOUREL, 2003), (BETTADAPURA, 2012), (FASEL; LUETTIN, 2003). A tarefa de análise de uma face, necessária para o reconhecimento posterior de uma expressão facial, pode ser dividida em dois grupos de abordagem: métodos holísticos e métodos locais. Na visão holística a face é vista pelo sistema como um todo, ou seja, é a entrada única composta por todos os elementos da face como testa, olhos, sobrancelha e nariz. O segundo grupo é composto por técnicas locais onde a face é dividida em sub regiões de interesse, tais como os olhos, nariz, boca. Nesse caso, a análise da expressão é realizada segundo resultados da análise local de cada elemento. A Figura 1.4 ilustra as duas abordagens.



(a) (b) (c)

Figura 1.6: A imagem original em (a) é analisada de forma holística em (b) e de forma local em (c).

Para a análise local pode-se ter componentes permanentes, formadas pelas características

locais que sempre estão presente na imagem como olhos e boca ou por componentes transientes, formadas por características esporádicas como enrugamento de testa ou avermelhamento da pele.

A Tabela 1.1 sumariza alguns métodos utilizados na literatura para o problema de identificação de expressão facial. Destacam-se para a composição dos descritores de característica, isto é, métodos que extraem dados de uma face para posterior entrada no bloco de classificação, as técnicas geométricas, como utilização da distância Euclidiana e Grafos (*Labeled Graph*-(LG)), Fluxo Óptico, Imagem Diferença, Filtro de Gradiente, *Wavelet* de Gabor, Detectores de Canny, *Principal Component Analysis* - PCA (*eigenface*), *Linear Discriminant Analysis* - LDA, *Independent Component Analysis* - ICA, *tracking* com *Piecewise Bézier Volume Deformation* - PBVD , Candide e Filtro de Partículas, *Higher Order Local Autocorrelation* - HLAC, Descritores *Topographic Context* - TC, *Raw Pixels*, *Local Binary Patterns* - LBP, *Directional Ternary Pattern* - DTP, *Active Appearance Model* - AAM, dentre outros.

Para a etapa de classificação da expressão facial destacam-se as técnicas de redes neurais (*Neural Networks* - NN) como o *Multilayer Perceptron*- MLP, *Radial Basis Function Neural Network*, PCA, LDA (Mapa de Fisher), ICA, modelos estatísticos como o *Naive Bayes* - NB, *Hidden Markov Model* - HMM, *Quadratic Density Classifier* - QDC e o Gini *Index*, correlação com *Kernel Canonical Correlation Analysis* - KCAA , Transformada de Curvlet, representação esparça (*Sparse Representation Classifier* - SRC) e a Máquina de Vetores de Suporte ou *Support Vector Machine* - SVM.

Nesse trabalho, optou-se pelo modelo estatístico AAM devido a sua grande versatilidade para extração de características. O modelo gerado pode ser utilizado em várias aplicações. Os parâmetros que determinam a forma e textura de uma expressão facial podem compor o vetor de características utilizado na entrada de um classificador. No entanto, outras áreas podem ser exploradas. A partir de um banco de dados é possível criar uma série de faces e expressões sintéticas utilizando os parâmetros obtidos durante a fase de treinamento estatístico. Outra aplicação do AAM é na localização de pontos que compõem a face. A localização do centro dos olhos, por exemplo, é de extrema importância em certas aplicações como no comando remoto utilizando os movimentos dos olhos. O AAM é capaz de gerar uma estimativa estatística dessa localização, apresentando uma solução alternativa a métodos de verificação local utilizando filtros e operadores morfológicos. Para a etapa de classificação utilizou-se o modelo estatístico K-NN e, posteriormente, a máquina de vetores de suporte (SVM).

Salienta-se que o trabalho apresentando realiza a localização das faces utilizando o *framework* Viola-Jones, diferentemente dos trabalhos citados que localizam a face manualmente ou as

Referência	Extrator de Características	Classificador
(ZHANG et al., 1998)	Geometria (*Labeled graphs*); *Wavelet* de Gabor	MLP
(LIEN et al., 1998)	Filtro de Gradiente	HMM
(LYONS; BUDYNEK; AKAMATSU, 1999)	*Wavelet* de Gabor	LDA
(DONATO et al., 1999)	Imagem Diferença; Fluxo Óptico; *Wavelet* de Gabor	PCA; ICA
(COOTES; TAYLOR, 2004)	AAM	-
(TIAN; KANADE; COHN, 2001)	Fluxo Óptico; *Wavelets* de Gabor; Detectores de Canny	NN
(DUBUISSON; DAVOINE; MASSON, 2002)	PCA *eigenface*; LCA	Subespaços Discriminantes; Distância Euclidiana
(BUCIU; KOTROPOULOS; PITAS, 2003)	Filtro de Gabor; ICA	CSM; SVM
(BARTLETT et al., 2003)	*Wavelet* de Gabor	SVM; *Adaboost* SVM
(COHEN et al., 2003)	PBVD *tracker*	SVM
(MICHEL; KALIOUBY, 2003)	Geometria: Distância Euclidiana	SVM
(SHINOHARA; OTSU, 2004)	HLAC;	Mapa de Fisher
(PANTIC; ROTHKRANTZ, 2004)	Geometria: Perfil pontos frontais	*Rule based*
(PANTIC; PATRAS, 2005)	*Tracking* pontos faciais fiduidais	Regras Temporais
(ZHENG et al., 2006)	*Labeled Graph*-(LG)	Correlação com KCCA
(PANTIC; PATRAS, 2006)	*Tracking* com Filtro de Partículas	*Rule based*
(KOTSIA; PITAS, 2007)	Candide *nodes*	SVM
(WANG; YIN, 2007)	Descritores do tipo *Topographic context*-(TC)	QDC; LDA; SVC; NB
(KOTSIA; BUCIU; PITAS, 2008)	Descritores de Gabor; DNMF; Geometria com Candide *tracker*	SVM; MLP
(SHIH; CHUANG; WANG, 2008)	Filtro de Gabor; PCA; ICA	LDA; SVM
(MARTINS; SAMPAIO; BATISTA, 2008)	AAM	SVM
(LAJEVARDI; HUSSAIN, 2009)	HLAC; LBP	*Curvelet Transform*; SVM
(CHANG; HUANG, 2010)	Descritos Faciais baseados na Identidade do Indivíduo;	RBF-NN
(SONG; CHEN, 2011)	AAM;	BPNN
(ZHANG; ZHAO; LEI, 2012)	*Raw pixels*; LBP; *Wavelet* de Gabor	SRC; NN; SVM
(AHMED; KABIR, 2012)	*Directional Ternary Pattern*-DTP	-
(PERVEEN; GUPTA; VERMA, 2012)	Geometria (Pontos da Face)	Gini *Index*

Tabela 1.1: Técnicas comuns na literatura para reconhecimento de expressões faciais.

tratam já previamente recortadas.

1.5 Sistema Proposto

O sistema proposto deve ser capaz de realizar o reconhecimento de uma expressão facial de maneira automática, ou seja, sem executar nenhum pré-processamento nas imagens de entrada.

Nesse trabalho é realizada a localização de faces utilizando o *framework* Viola-Jones baseado no *Adaboost*, a extração de características aplicando o AAM e a classificação empregando o k-NN e o SVM com *kernel* RBF conforme Figura 1.7.

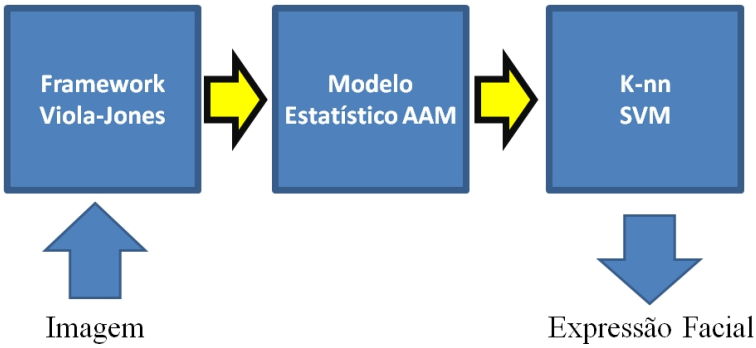


Figura 1.7: Diagrama de blocos proposto nesse trabalho: Detecção de Face utilizando o *framework* Viola-Jones, extração de características através do AAM e classificação com k-NN e SVM.

O trabalho foi elaborado no Laboratório Computadores e Redes Neurais - CISNE, pertencente ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal do Espírito Santo - PPGEE UFES.

Os algoritmos foram desenvolvidos utilizando o Matlab™ como base. Adicionalmente foram utilizados pacotes do OpenCV para chamada do Viola-Jones e o *toolbox* svmlib (CHANG; LIN, 2011) para classificação.

1.6 Organização do Trabalho

Este trabalho está organizado da seguinte maneira:

- Capítulo 1: É apresentado o problema, sua relevância e caracterização. Na sequência são delineados os objetivos do trabalho, a metodologia empregada e o sistema proposto.

- Capítulo 2: O problema de localização da face é exposto e emprega-se o *framework* Viola-Jones para extrair as faces do banco de dados.

- Capítulo 3: As características que compõem a expressão facial e devem ser classificadas são extraídas utilizando modelagem estatística do algoritmo *Active Appearance Model* - AAM, onde são gerados modelos de formato, textura e aparência. Nesse procedimento também é realizada uma redução de dimensionalidade.

- Capítulo 4: Os dados extraídos na etapa anterior devem ser classificados e são apresentados o classificador paramétrico *k-Nearest Neighbor* k-NN (ou k Vizinhos Mais Próximos) e o método de aprendizagem supervisionada *Support Vector Machine* - SVM (ou Máquina de Vetores de Suporte) com *kernel Radial Basis Function* - RBF e a utilização de *kernels* para tratar problemas não lineares.

- Capítulo 5: Esse capítulo é destinado a apresentação do banco de dados e resultados obtidos na localização de face, extração de características e classificação. É proposto um conjunto de pontos para inicialização da busca do AAM e uma nova atualização dos pesos no algoritmo de busca das características, minimizando os erros entre o modelo sintético gerado e a imagem original.

- Capítulo 6: Apresenta as conclusões, comparações com trabalhos pertinentes e sugestões de trabalhos futuros.

Capítulo 2

Detecção de Faces

Em um sistema de identificação de expressões faciais a etapa de classificação depende de uma boa extração de características para um bom desempenho, tarefa que só é possível mediante a localização precisa da face. Portanto, a taxa de acerto do classificador estará diretamente relacionada com a taxa de detecção de face.

A detecção de face deve ser insensível a fatores como etnicidade, idade, gênero, pelos do indivíduo, posição da face, iluminação, resolução, escala, contraste, níveis de cor e complexidade de fundo de cena (*background*), conforme Figura 2.1. Oclusão do alvo e utilização de acessórios como óculos, bonés e maquiagem são desafios adicionais.

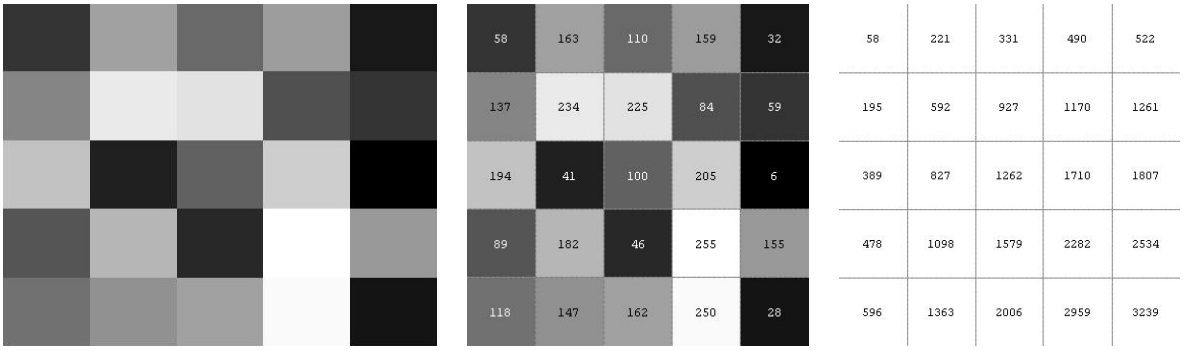
(a) (b) (c)

Figura 2.1: Fatores como (a) etnicidade, idade, gênero, pelos do indivíduo, (c) complexidade de cena, escala, resolução, (c) uso acessórios, iluminação e contraste são complicadores para o problema de detecção de face.

Portanto, é necessário um método robusto para esse módulo. O *framework* de detecção de objetos Viola-Jones (VIOLA; JONES, 2001) é amplamente utilizado na área, apresentando bons resultados e é capaz de trabalhar em tempo real, sendo brevemente descrito a seguir.

2.1 *Framework* Viola-Jones

O algoritmo inicializa-se calculando a imagem integral, onde cada *pixel* $P(x, y)$ é substituído pela somatória dos valores de todos os *pixels* localizados acima e à esquerda de $P(x, y)$, conforme Figura 2.2. Dessa forma, é possível calcular o valor da integral de uma região D delimitada pelos *pixels* ($P1$, $P2$, $P3$ e $P4$) a partir da imagem integral rapidamente fazendo: $P4 - P3 - P2 + P1$.



(a) (b) (c)

Figura 2.2: (a) Imagem teste de 5x5 *pixels*, (b) os valores de cada *pixel* e (c) a imagem integral.

Esse procedimento é de extrema utilidade porque a detecção do objeto é realizada através da análise de características de regiões retangulares na imagem, havendo similaridades com as funções base de Haar utilizada na Transformada *Wavelet* de *Haar* (GONZALEZ; WOODS, 2007), conforme Figura 2.3. Existem três tipos de características que podem ser exploradas:

- característica de dois retângulos: diferença entre a soma de *pixels* entre duas regiões retangulares;
- característica de três retângulos: diferença entre a soma de *pixels* de duas regiões retangulares laterais e uma região central;
- característica de quatro retângulos: diferença entre a soma de *pixels* entre duas regiões retangulares de uma diagonal e a soma de *pixels* das duas regiões retangulares na diagonal oposta.

A limitação obtida pela simplicidade de uma região retangular ao invés de uma região mais complexa é compensada pela robustez computacional, característica marcante do algoritmo de Viola-Jones

Figura 2.3: O *Adaboost* utiliza um conjunto de classificadores fracos para determinar um classificador forte.

O processo de treinamento do classificador que determina quais regiões são potenciais alvos, no caso particular desse trabalho as regiões de faces, é realizado através de estímulos com exemplos positivos (imagens com faces) e exemplos negativos (imagens sem face). Apesar de existir uma variedade de possibilidades de características, apenas poucas são suficientes para determinar a face. Viola-Jones utiliza uma variação do *AdaBoost* (ver Apendice B) tanto para selecionar as características relevantes quanto para treinar o classificador, conforme apresentado na Figura 2.4. O algoritmo *Adaboost* é utilizado para melhorar o desempenho de um classificador através da combinação de funções de classificação fracas para formar um classificador mais forte. Por exemplo, após o processo de treinamento de um classificador fraco ser encerrado, é realizado uma atribuição de pesos aos exemplos na entrada de um novo classificador com intenção de enfatizar os exemplos que foram incorretamente classificados a princípio.

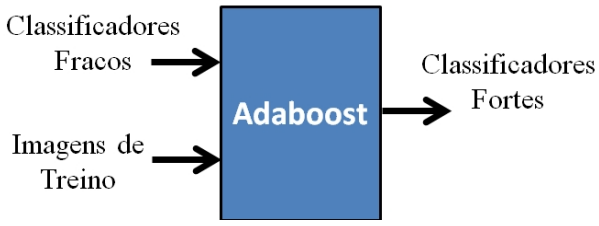


Figura 2.4: Exemplos de características candidatas a serem utilizadas no classificador.

A convergência dos classificadores fracos para o classificador forte é utilizada para obter o menor número de erro de classificação. A cada iteração do *Adaboost* são atribuídos pesos maiores para algumas características que são mais representativas para os exemplos de entrada (faces). Como consequência, obtém-se um classificador com poucas características e capaz de detectar o objeto de interesse. O erro decai de forma exponencial a cada iteração e utilizando uma margem maior de exemplos ocorre uma maior generalização. Na prática, é

escolhida a região retangular que melhor separa os exemplos negativos dos positivos (escolha das características que serão usadas no classificador de face).

O classificador final do Viola-Jones é obtido como a combinação linear de alguns classificadores associados a diferentes características (regiões retangulares) e o peso de cada classificador é inversamente proporcional à taxa de erro obtida. O procedimento de seleção das características é alcançado utilizando alguma regra de aprendizagem como o Winnow exponencial (KIVINEN; WARMUTH; AUERC, 1997). Forma-se um conjunto de características onde cada *pixel* é mapeado em um vetor binário f com dimensão d, $[0, d-1]$. Quando um *pixel* assume o valor k, a dimensão k assume valor 1 enquanto as outras dimensões assumem valor 0: $f_k = [0\ 0\ 0\ ...\ 0\ 1\ 0\ ...\ 0]$. Os vetores são concatenados, fazendo-se $F = [f_0\ f_1\ f_2\ ...\ f_n]$, formando um vetor esparso de dimensão nd, onde n é o número de *pixels*. A regra de classificação converge para muitos dos pesos tornarem-se iguais a zero. Observa-se, portanto, que há um interesse em se trabalhar com valores de intensidade/luminosidade do *pixel*/área.

Para a detecção de faces, a saída do algoritmo de aprendizagem de seleção de características escolhe regiões centradas nos olhos. A primeira é a região que compreende olhos e a parte abaixo dos olhos. Isso se deve ao fato de que regiões dos olhos são normalmente mais escuras que a região da bochecha. A segunda característica envolve os olhos e a ponta do nariz, pois os olhos são mais escuros que a ponta do nariz, em geral. Portanto, é coerente afirmar que o algoritmo de Viola-Jones para o caso de detecção de faces procura, a princípio, os olhos.

Nos testes realizados por Viola-Jones foi alcançado uma taxa de acerto de 95% para o banco de testes com 1 falso positivo para cada 14.084 imagens. A varredura da imagem começa com sub-janelas de 24×24 *pixels* e percorre-se toda a imagem. Em seguida, as janelas são aumentadas por um fator de 1,25 e o procedimento se repete até o tamanho da janela atingir o tamanho total da imagem. Observa-se que se faz necessário um procedimento de converter várias entradas de face localizadas para a mesma face em diferentes resoluções de janela para apenas uma face e uma localização.

Apesar da redução do número de características de Haar obtidas com o treinamento do classificador selecionador de características *Adaboost*, tal como foi apresentado o algoritmo, ainda seria difícil trabalhar em tempo real com algumas centenas de características em várias escalas devido ao custo computacional. Assim, Viola e Jones propuseram a utilização de uma estrutura de árvore de decisão onde há vários classificadores em cascata (*Attentional Cascade*) para reduzir o tempo de processamento e, ainda com uma melhora na taxa de acerto. A chave da árvore de decisão é construir um classificador capaz de maximizar a rejeição das imagens negativas (que não contém face) e simultaneamente capaz de detectar a maioria das imagens positivas (com face). Classificadores são cascateados de forma que a

saída classificada como face é a entrada de um classificador mais complexo visando diminuir a taxa de falso positivo. Verificou-se que utilizando um classificador com apenas as duas características principais na ponta da cascata é possível reconhecer 100% das faces com uma taxa de falso positivo de 40%, o que representa um elevado ganho computacional, uma vez que a imagem só é processada nos demais estágios da cascata ao passar com sucesso pelo estágio antecessor. Um classificador construído com duas características utiliza apenas cerca de 60 micro instruções para decidir se a imagem é positiva ou negativa.Dessa forma, a maioria das imagens negativas é rapidamente descartada e evita-se processamentos posteriores.

A face só é detectada se todos os estágios em cascata derem positivos. O algoritmo pára ao primeiro caso negativo. Observa-se que ao deslocarmos as janelas ao longo das imagens, e em várias escalas, é esperado que a maioria das saídas de cada janela sejam negativas. Logo, esse processo em cascata torna possível a implementação do algoritmo em tempo real e possibilita maior tempo de processamento em etapas futuras como, por exemplo, na extração de características da face e na classificação das emoções associadas.

A aceitação ou rejeição de uma face em uma imagem no *framework* Viola-Jones é determinada pelo nível de coincidência entre os dados apresentados na entrada e os dados previamente treinados. Portanto, se faz necessário determinar um limiar de decisão. Esse limiar pode gerar identificação de faces que não constam na base de dados e rejeição de faces que deveriam ser aceitas pelo sistema. Portanto, assim como todo sistema, existem níveis de falhas que devem ser tratados observando a relação de compromisso entre resultado e segurança que deve existir para garantir um bom desempenho do sistema dentro de suas propostas com o mínimo de falhas possível.

Quando uma face é encontrada em uma imagem que não contém face ocorre um erro associado à FAR- *False Acceptance Rate*, ou taxa de falsa aceitação, que é dada pela relação entre o número de imagens que não são faces que são aceitas pelo sistema e a quantidade total de imagens que não são faces. Na Figura 2.5 pode ser observado que o nível de falsas aceitações aumenta quando é diminuído o limiar do sistema para FAR.

$$FAR = \frac{\text{Entradas Erroneamente Aceitas}}{\text{Total de Entradas do Sistema}} \tag{2.1}$$

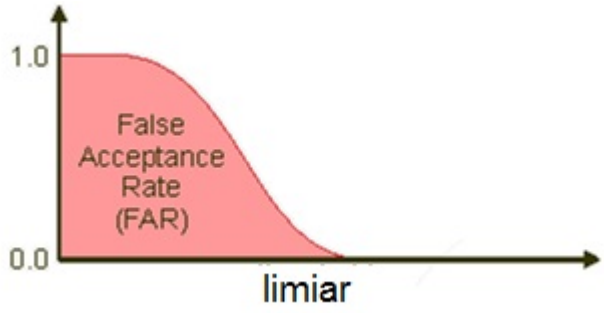


Figura 2.5: Limiar para FAR- *False Acceptance Rate*.

Outra possibilidade de erro ocorre quando uma imagem que deveria conseguir ser classificada como face é rejeitada, ou seja, é encarado como um exemplo negativo. Esse erro é causado quando a imagem apresenta um nível de coincidência de suas características coletadas abaixo do esperado pelo sistema treinado. Esse erro é associado à FRR - *False Rejection Rate*, ou taxa de falsa rejeição, é a relação entre o número de faces classificadas como não face ao serem inseridas no sistema e o número total de faces apresentadas ao sistema. Na Figura 2.6 pode ser observado que o nível de falsas rejeições aumenta ao aumentarmos o limiar do sistema para FRR.

$$FRR = \frac{\text{Entradas Erroneamente Rejeitadas}}{\text{Total de Entradas do Sistema}} \tag{2.2}$$

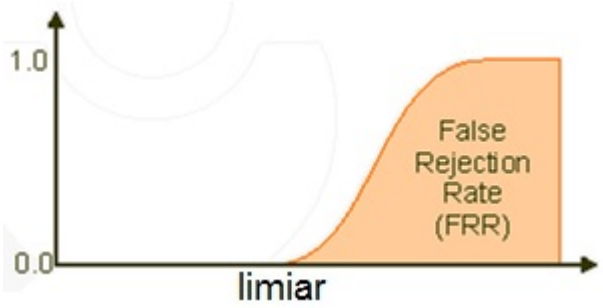


Figura 2.6: Limiar para FRR - *False Rejection Rate*.

Analisando a FAR e a FRR temos que estabelecer um limiar que minimize os possíveis erros causados na utilização do sistema. Estamos diante de uma relação de compromisso, onde aumentar o limiar do sistema representa aumentar a FRR e diminuir o limiar representa aumentar a FAR. Uma possível escolha para o limiar está na interseção das curvas da FAR e FRR. Esse ponto é conhecido como EER - *Equal Error Rate*, ou taxa igual de erro, conforme Figura 2.7. Esse limiar é típico em aplicações de uso doméstico, onde há um equilíbrio entre

a segurança e a comodidade do usuário, como em acesso à sistemas biométricos.

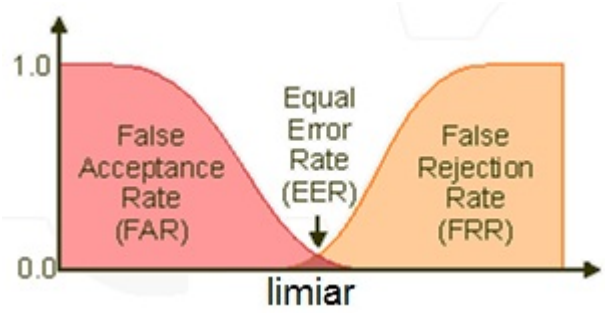


Figura 2.7: O limiar do sistema relaciona FAR e FRR.

Para o *framework* Viola-Jones, é interessante observar que cada estágio possui uma FAR e uma FRR. Os classificadores de cada estágio são treinados com os exemplos negativos que passaram pelo estágio anterior (falso positivo). A estrutura em cascata também apresenta uma propriedade interessante: a taxa de falso positivo F e a taxa de detecção D é obtida com o produto das taxas individuais de falsos positivos f_i e taxas individuais de detecção d_i de cada um dos K estágios, como expresso em

$$F = \prod_{i=1}^{K} f_i, \tag{2.3}$$

$$D = \prod_{i=1}^{K} d_i. \tag{2.4}$$

Portanto, um classificador com 90% de taxa acerto pode ser construído utilizando 10 estágios com $d_i = 99\%$, totalizando $D = 0.99^{10} = 0.9$. Essa tarefa é facilitada pelo fato de se alcançar 99% de detecção em cada estágio pagando-se o preço de obter 30% de falsos positivos, um valor elevado por estágio, mas reduzido na saída do classificador em cascata, pois $F = 0.3^{10} \approx 6 \times 10^{-6}$).

Observa-se que o *Adaboost* tem como característica a tentativa de minimizar o erro de classificação de faces ao passo que é desejável um classificador que procure uma relação de custo x benefício com alta taxa de acerto mesmo com altas taxas de falso positivo. Uma maneira de alterar esses parâmetros é ajustar o limiar do perceptron (BISHOP, 2006), responsável pela seleção de características no *Adaboost*. À medida que se aumenta o limiar reduz-se o número de falso positivos e taxa de acerto. Limiares menores resultam em maiores taxas de detecção.

É necessário definir uma relação entre o número de classificadores/estágios, o número de características utilizadas em cada classificador/estágio e o limiar de cada estágio. Essa tarefa é complicada, mas tem uma solução prática simples: define-se um limiar para os verdadeiros positivos (taxa de detecção) e para a taxa de falso positivo. Uma nova característica é acrescentada no estágio em cada iteração até que sejam alcançados os valores que respeitem os limiares estabelecidos.

A estrutura final do detector em cascata utiliza 32 camadas/estágios com um total de 4.297 características (Figura 2.8).

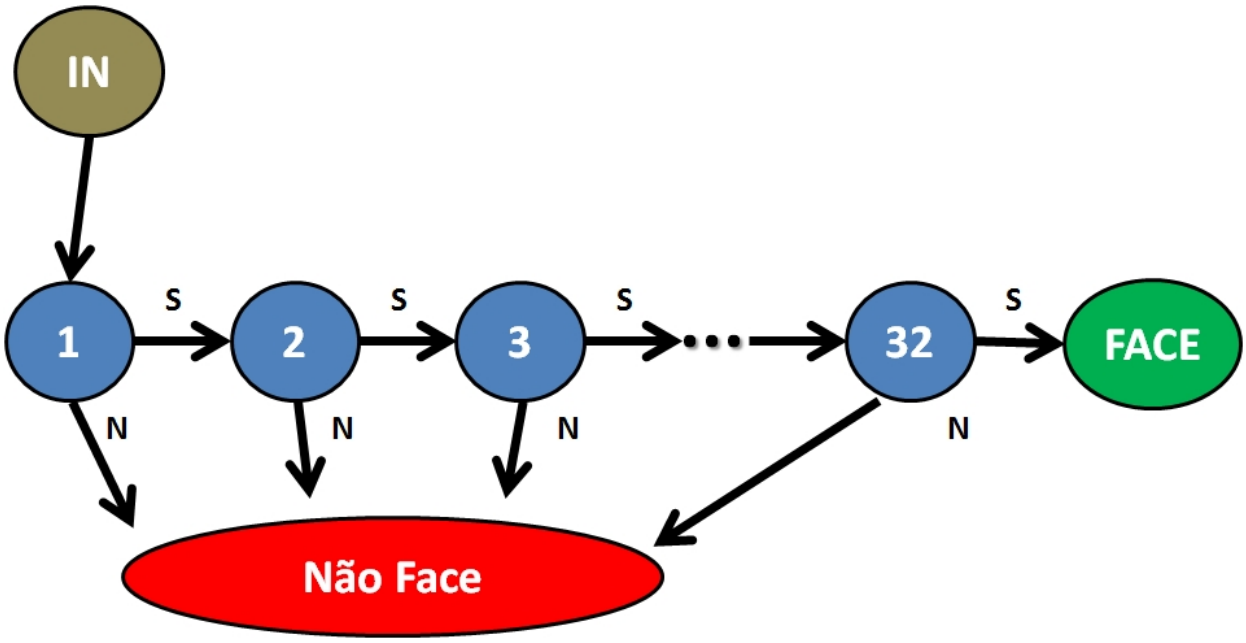


Figura 2.8: Estrutura final do classificador com 32 estágios e 4.297 características.

Capítulo 3

Modelo de Aparência Ativa: AAM - *Active Appearence Model*

3.1 Modelagem Estatística

Uma variável aleatória x é uma função que atribui um resultado numérico a cada resultado individual de uma experiência. Se analisarmos os seres humanos verificamos que suas faces são formadas por elementos básicos como olhos, nariz e boca, independente de sua configuração. Ou seja, identificamos olhos em todos os humanos apesar de cada indivíduo apresentar um formato e textura para os olhos: uns mais esticados, outros mais arredondados e cada um com cor diferente. Portanto, cada elemento da face pode ser encarado como uma variável aleatória e, por consequência, a própria face é uma variável aleatória bem como as expressões faciais que a compõem.

Podemos delimitar todas as expressões faciais possíveis no conjunto discreto formado pelas expressões faciais básicas de alegria, raiva, medo, tristeza, surpresa, nojo/aversão, desprezo e neutra. Dessa forma, podemos trabalhar com um conjunto discreto de valores e modelar estatisticamente as faces e/ou expressões faciais.

Para modelagem da expressão facial o primeiro passo é localizar a face. Em seguida, é necessário extrair as características que compõem a expressão facial. O AAM[1]- *Active Appearance Model* (COOTES; EDWARDS; TAYLOR, 1998) é um algoritmo capaz de modelar estatisticamente um objeto baseado na decomposição de forma e textura que compõem um conjunto de protótipos de treino similares submetidos a um treinamento. Além de modelar

uma face e, consequentemente, sua expressão facial, o AAM gera redução de dimensionalidade, diminuindo o custo computacional de operações que necessitem representações da face em etapas futuras.

Para aplicar o modelo AAM é necessário um conjunto de amostras/protótipos que representem a variável aleatória a ser modelada. No caso de faces o conjunto de treino é formado por imagens de faces contendo diferentes expressões faciais. A delimitação da região da face e plano de fundo é realizada por um conjunto de pontos ou marcações que constituem o contorno da região que contém os componentes de interesse da face, englobando olhos, nariz, boca, sobrancelhas, queixo e bochecha. Esse conjunto de pontos são conhecidos como *landmarks* e todas as imagens do banco de dados devem ser capazes de terem esses pontos extraídos.

Com a delimitação da região de face e a obtenção dos *landmarks*, é possível gerar uma distribuição espacial para a variável aleatória obtida com os pontos do banco de dados e gerar um modelo de forma para a face. Com os mesmos pontos é possível mapear regiões, através de triangulação e utilizar os *pixels* delimitados por essas regiões como outra variável aleatória e gerar um modelo de textura.

A formação do banco de dados é, portanto, fundamental para o sucesso da modelagem. As imagens de entrada devem conter apenas faces, o que demanda um bom pré-processamento no bloco de localização de faces, e as faces modeladas devem ser representativas em relação às faces que não pertencem ao banco e serão utilizadas para testes. Ou seja, se no banco há apenas imagens com bocas fechadas, não é esperado uma boa modelagem para imagens com bocas abertas.

3.2 Modelo Estatístico da Forma

Uma vez estabelecido um banco de dados, o primeiro modelo gerado pelo AAM é o modelo de forma. Como não há uma padronização nas imagens de entradas em relação à posição ou inclinação da face, as s imagens do banco de treino podem estar em escalas e orientações diferentes, sendo necessário um procedimento de alinhamento dos pontos entre cada forma (*landmark*) x dos s exemplos de treino para gerar o modelo para cada imagem i. Como cada ponto é formada por duas coordenadas, teremos um conjunto de $2n$ pontos, onde n é o

[1] O AAM - *Active Appearance Model* pode ser traduzido para o português como Modelo de Aparência Ativa , onde aparência está associada à composição de forma em conjunto com a textura. Nesse trabalho será mantida a sigla em inglês.

número de pontos no *landmark*. Se considerarmos que o formato de um objeto é invariante a transformações Euclidianas, podemos alinhar cada vetor de treino reposicionando-os, escalonando e rotacionando-os. É realizada a Análise de Procrustes(STEGMANN; GOMEZ, 2002), reduzindo os efeitos de escala, rotação e translação com os seguintes passos:

- translada-se cada exemplo de treino de forma que seu centro de massa coincida com a origem;
- define-se um dos exemplos de treino como sendo uma estimativa inicial da forma média e normaliza-se o vetor de forma que $||\bar{x}|| = 1$.
- alinha-se todos os exemplos de treino de acordo com a estimativa para forma média;
- incrementa-se i e estima-se um novo vetor de forma média $||\bar{x}^{i+1}||$ utilizando os exemplos alinhados no passo anterior. Normaliza-se $||\bar{x}^{i+1}||$;
- avalia-se a equação de custo $D = \sum_{i=1}^{s} ||x^i - \bar{x}||$, objetivando minimizar a distância entre cada exemplo de treino do vetor de forma média $\bar{x}$. Verifica-se se houve convergência. Caso positivo, a análise termina. Caso contrário, retorna-se ao terceiro passo até obter convergência.

O procedimento fornece uma medida estatística da distribuição de cada ponto de entrada. O próximo passo é equacionar um modelo M com parâmetros b que represente a diferença entre os exemplos de treino e a forma média de treinamento na forma $x = M(b)$. O vetor b é denominado vetor de forma. A dimensão máxima de b será $2n$, utilizando todos os pontos do *landmark*. Como é desejável utilizar no modelo apenas componentes significativas para o modelo, é adequado a utilização de Análise de Componentes Principais - PCA para redução de dimensionalidade (ver Apêndice A). Nesse caso, o parâmetro b assume uma dimensão $b \leq 2n$. O modelo é obtido fazendo-se os passos:

- computa-se o vetor de forma média

$$\bar{x} = \frac{1}{s} \sum_{i=1}^{s} x^i, \tag{3.1}$$

- calcula-se a covariância amostral

$$S = \frac{1}{s-1} \sum_{i=1}^{s} (x^i - \bar{x})(x^i - \bar{x})^T, \tag{3.2}$$

- calcula-se os autovetores Φ_i e os autovalores $\lambda_{s,i}$ associados à S. Ordena-se os autovalores de forma decrescente, ou seja, $\lambda_{s,i} \geq \lambda_{s,i+1}$.

Portanto, se considerarmos os z autovetores associados aos z maiores autovalores, para $z \leq 2n$, podemos reduzir a dimensionalidade e aproximar qualquer exemplo como

$$x \approx \bar{x} + P_s b_s, \tag{3.3}$$

onde $P_s = [\Phi_1|\Phi_2|...|\Phi_z]$ e b_s é o parâmetro de forma e representa b após redução de dimensionalidade. Ou seja, P_s é o novo espaço formado pelos principais autovetores dos dados de forma e b é responsável pelo peso atribuído a cada autovetor, gerando diferentes formas no novo espaço.

Portanto, uma forma qualquer pode ser representada pela forma média mais uma deformação gerada pela associação dos autovetores ponderados pelo parâmetro b_s. Os parâmetros do modelo M podem ser definidos através da matriz P_s e a forma média, fazendo :

$$b_s^i = P_s^T(x^i - \bar{x}). \tag{3.4}$$

Portanto, é possível descrever uma expressão facial através da disposição de seus pontos segundo o modelo treinado. Essa etapa é denominada ASM - *Active Shape Model*(COOTES; EDWARDS; TAYLOR, 1998).

No entanto, mais informações podem ser obtidas se levarmos em consideração a textura da imagem que contém informações importantes como sombras e enrugamento nas regiões dos olhos, discriminantes na detecção de uma expressão facial.

3.3 Modelo Estatístico de Textura

A textura de uma imagem é dada pela maneira como os *pixels* estão dispostos em uma determinada região ou caminho formando um padrão de cores ou intensidades e, portanto, contém informações descritivas e discriminativas que podem auxiliar na construção de um modelo mais robusto de representação além da forma. O modelo estatístico exclusivamente

de textura deve ser independente das formas dos protótipos de treino. Para tal, é realizada uma deformação em cada imagem de maneira que seus pontos referentes à forma coincidam com o modelo de forma média. No caso bidimensional podemos utilizar a triangulação de Delaunay para delimitar as áreas que compõem a textura da face utilizando os *landmarks* como referência.

Em seguida, é realizado uma transformação em cada protótipo de treino onde cada pixel original no interior do triângulo formado por 3 pontos que compõem a forma original será mapeado em um novo pixel em uma imagem deformada para a forma média de acordo com a posição de cada vértice dos triângulos.

Ou seja, para o modelo de textura a informação de formato deve ser descartada fazendo um remapeamento através dos pontos delimitados por Delaunay onde cada face de entrada é levada à forma média. Portanto, o modelo gerado será o modelo de textura vinculado à forma média.

Ao escolher as imagens de treino que compõem o banco de dados pode-se obter imagens com diferentes condições de iluminação, interferindo na análise estatística da textura. Regiões com reflexos ou pouco iluminadas, por exemplo, distorcem os valores reais dos *pixels* que compõem a cena, elevando ou diminuindo excessivamente seus valores. Nesse caso, há a necessidade de se alinhar fotometricamente as texturas de treinamento para minimizar os efeitos indesejados introduzidos pelas diferentes condições de iluminação de cada imagem de entrada.

O processo de alinhamento é realizado normalizando as texturas conforme

$$u : g^i = \frac{g^i - \bar{g}^i}{\sigma^i}, \tag{3.5}$$

onde g^i e $\bar{g}^i$ são a intensidade e a média das intensidades dos pixel da textura, respectivamente, e σ^i é o desvio padrão para as intensidades dos *pixels* imagem de entrada i.

Similar ao modelo de forma, a distância Euclidiana E_g entre cada vetor de textura de treino e a textura média é minimizada segundo a expressão:

$$E_g = \sum_{i=1}^{s} ||g^i - \bar{g}||, \tag{3.6}$$

onde $\bar{g}$ é a textura média para o conjunto de treino.

O modelo de textura é inicializado utilizando uma das amostras de treino como referência. Através de iterações sucessivas controladas pelo índice de cada protótipo i, como no modelo de forma, deseja-se minimizar E_g, resultando em um vetor de textura $\bar{g}$ médio para o modelo, após convergência.

Novamente, aplicando-se PCA, podemos representar uma textura em função de parâmetros de textura b_g como $g \approx \bar{g} + P_g b_g$, onde P_g contém os k autovetores correspondentes aos maiores k autovalores λ_g.

Portanto, considerando os dois modelos, podemos definir uma imagem de entrada em termos de b_s e b_g. O treinamento de forma fornece os vetores médios de forma $\bar{x}$ e a matriz P_s, conforme Equação 3.4. Já o modelo de textura apresenta $\bar{g}$ e as matriz P_g, descrita por

$$b_g = P_g^T(g - \bar{g}). \tag{3.7}$$

Ou seja, a textura de qualquer imagem no formato médio pode ser representada como a textura média do modelo somada aos autovetores ponderados pelo vetor de textura b_g.

3.4 Modelo Estatístico Combinado

Espera-se que modelos estatísticos de forma e textura não sejam independentes, uma vez que ao mudar a textura de uma face é necessário alterar algum parâmetro local como abertura de boca ou olhos e, portanto, altera-se a forma ou posicionamento dos *landmarks*. Alternativamente, alterar o formato de uma face introduz mudanças nas texturas. Um modelo que una informações de textura e forma, minimizando redundância de informação é definido como modelo combinado ou modelo de aparência, aqui denominado por c.

O modelo de aparência utiliza o parâmetro de forma b_s e o parâmetro de textura b_g, combinando as informações em um único parâmetro de aparência b_{sg}. No entanto, b_s está na forma de unidades de distância (refere-se ao posicionamento do *pixel*), ao passo que b_g possui unidade de intensidade de *pixel*. A relação entre os dois parâmetros é realizada atribuindo uma matriz de pesos W_s ao parâmetro responsável pela forma, relacionando o efeito que alterar b_s gera nas intensidades de *pixels* relacionadas à b_g, conforme observa-se na expressão:

$$b_{sg} = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix}. \tag{3.8}$$

Portanto, W_s é calculado variando-se a forma de um exemplo do banco de dados através de um ajuste no valor de b_s^i e observando o efeito resultante na intensidade dos *pixels* que são refletidos na textura e, consequentemente, nos valores de b_g^i. Uma média obtida para cada um dos W_s^i resultantes é utilizada como matriz de peso. Ou seja,

$$W_s = \frac{1}{N} \sum_{i=0}^{N} W_s^i. \tag{3.9}$$

W_s é uma matriz diagonal, onde cada elemento relaciona os pesos atribuídos aos autovetores de P_s que devem ser alinhados com P_g no modelo combinado. Logo, uma outra possibilidade para estimar W_s com menor custo computacional é .

$$W_s = \begin{bmatrix} W_{11} & 0 & \cdots & 0 \\ 0 & W_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Wz,k \end{bmatrix}, \tag{3.10}$$

onde

$$W_{ij} = \frac{\sum_{i=1}^{z} \lambda_{s,i}}{\sum_{i=1}^{k} \lambda_{g,j}}. \tag{3.11}$$

O modelo final é obtido aplicando-se a análise de componentes principais em b_{sg}:

$$b_{sg} \approx P_c c. \tag{3.12}$$

onde P_c é a matriz de autovetores associada aos primeiros m maiores autovalores. Dessa forma, temos uma redução de dimensionalidade $m \leq z + k$, onde z e k estão associados ao números de autovetores obtidos nos modelos de forma e textura, respectivamente.

A matriz P_c é obtida aplicando-se análise de componentes principais na matriz formada pelos parâmetros de textura b_g e pelos parâmetros de forma ponderados $W_s b_s$. Portanto, resultam uma parcela relacionada à textura P_{cg} e outra relacionada à forma P_{cs}, conforme Equação 3.13.

$$P_c = \begin{pmatrix} P_{cs} \\ P_{cg} \end{pmatrix}. \tag{3.13}$$

Observa-se que qualquer face pode ser representada por uma alteração de forma e textura, alterando-se b_s e b_g que, no modelo combinado, é representado, por definição, pelo vetor de aparência c. Portanto, qualquer face pode ser representada pelo modelo combinado e um vetor de aparência. Como o modelo é único, qualquer face é representada pelo vetor de aparência. Essa condição alimenta a utilização do vetor de aparência como entrada em um classificador de faces e/ou expressões faciais.

Uma imagem pode ser modelada em termos de seu formato e textura através das relações:

$$x = \bar{x} + P_s W_s^{-1} P_{cs} c \tag{3.14}$$

e

$$g = \bar{g} + P_g P_{cg} c, . \tag{3.15}$$

Ou, de maneira inversa, podemos sintetizar uma imagem através do vetor de aparência c.

3.5 Algoritmo de Busca

Dado que a partir do treinamento é possível parametrizar uma expressão facial por um vetor de aparência c, deseja-se encontrar uma expressão facial em uma nova imagem de entrada.

Qualquer imagem pode ser obtida a partir do modelo de aparência utilizando-se o modelo médio e a ponderação dos pesos dos autovetores associados ao vetor de aparência adequado. Portanto, aqui, estamos diante de um problema de otimização visto que é preciso, dado o

modelo desenvolvido, recuperar a expressão facial a partir de uma face de entrada.

O algoritmo de busca deve resolver o problema de diminuir a diferença δI entre a imagem real I_i e a imagem I_m sintetizada pelo algoritmo AAM, segundo a equação

$$\delta I = I_i - I_m. \tag{3.16}$$

Observa-se que δI possui implicitamente informações sobre a mudança necessária δc no vetor de aparência c que minimiza o erro entre a imagem de entrada e um modelo sintético testado que represente a mesma imagem. Portanto, a partir do conhecimento da relação entre δI e δc, é possível construir um algoritmo iterativo capaz de atualizar os valores do vetor de aparência e minimizar o erro entre o modelo e a imagem de entrada. Dado uma nova imagem de entrada x é possível criar uma versão sintética X a partir do modelo treinado através das transformações de escala S_x e S_y, rotação θ e translação l_x e l_y que são armazenadas em uma matriz de transformação em coordenadas homogêneas S_t segundo,

$$x : X = S_t(x). \tag{3.17}$$

Associado a essas transformações obtemos um vetor de pose $t = [S_x\ S_y\ l_x\ l_y]^T$, responsável por armazenar informações sobre transformações de translação, escala e rotação[2].

Define-se g^m como o modelo médio de textura e forma obtido durante a etapa de treinamento AAM, utilizando-o como estimativa inicial para qualquer nova face de entrada. O erro inicial é sempre a diferença entre o modelo treinado e a imagem de entrada amostrada dentro dos limites da região englobada pela forma média. Ou seja, o erro avalia a intensidade dos *pixels* da imagem de entrada amostrada nos limites definidos pela forma média do modelo. É realizada a normalização dos valores e é gerado um modelo de entrada g^s. O modelo inicial diverge das reais características da imagem e a diferença $r_{(p)}$ é dada por:

$$r_{(p)} = g^s - g^m, \tag{3.18}$$

onde

[2]Nesse texto x^T é a notação matemática para o vetor x transposto.

$$p = [c^T \; t^T]. \tag{3.19}$$

Portanto, a diferença $r_{(p)}$ é função do vetor de pose e aparência combinadas p. Ou seja, a diferença entre o modelo testado g^m e o modelo de entrada g^s da Equação 3.18 pode ser minimizada ajustando-se o vetor de aparência c e/ou no vetor de pose t, equivalente a ajustar os pesos das deformações introduzidas pelos autovetores treinados no AAM e ajustar a translação, escala e rotação do modelo testado.

Uma medida para a diferença é dada pelo somatório dos quadrados de cada elemento $E_{(p)} = r_{(p)}^T r_{(p)}$, ou seja, a Distância Euclidiana. Um novo treinamento deve ser realizado de forma a determinar qual deve ser a alteração δ_p nos limites de amostragem da imagem de entrada em uma nova iteração em função do erro prévio. Linearizando a variação $r_{(p+\delta_p)}$ pela expansão em Série de Taylor, truncada na primeira derivada, tem-se:

$$r_{(p+\delta_p)} \approx r_{(p)} + \frac{\partial r}{\partial p}\delta_p = r_{(p)} + \mathbf{J}\delta_p, \tag{3.20}$$

onde $\mathbf{J}$ é a matriz Jacobiana de dimensão $m \times n$ dada por

$$\mathbf{J} = \begin{pmatrix} \frac{\partial r_1}{\partial p_1} & \cdots & \frac{\partial r_1}{\partial p_n} \\ \vdots, & \ddots & \vdots \\ \frac{\partial r_m}{\partial p_1} & \cdots & \frac{\partial r_m}{\partial p_n} \end{pmatrix}, \tag{3.21}$$

onde m está relacionado à dimensão de r e n à dimensão de p.

Portanto, pode-se escrever o erro como

$$E_{(p)} = r_{(p)}^T r_{(p)} \tag{3.22}$$

$$E_{(p)} \approx (r_{(p)} + \mathbf{J}\delta_p)^T (r_{(p)} + \mathbf{J}\delta_p) \tag{3.23}$$

$$E_{(p)} \approx {\delta_p}^T \mathbf{J}^T \mathbf{J} \delta p + 2\delta_p^T \mathbf{J}^T + {r_{(p)}}^T. \tag{3.24}$$

Buscando-se o ponto onde a derivada do erro em relação a p é igual a zero obtém-se

$$0 = \mathbf{J}^T\mathbf{J}\delta p + \mathbf{J}^T r \tag{3.25}$$

$$\delta p = -(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T r \tag{3.26}$$

Dessa forma, a solução que minimiza o erro quadrático do problema é dada pela matriz de regressão R, como se observa a seguir.

$$\delta_p = -Rr_{(p)}, \tag{3.27}$$

onde

$$R = \left(\frac{\partial r}{\partial p}^T \frac{\partial r}{\partial p}\right)^{-1} \frac{\partial r}{\partial p}^T \tag{3.28}$$

$$R = \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T \tag{3.29}$$

$$R = \mathbf{J}^\dagger \tag{3.30}$$

Portanto, R é obtida utilizando a matriz pseudo inversa de $\mathbf{J}$ denotada por $\mathbf{J}^\dagger$.

Seria necessário o cálculo de R para cada imagem de entrada e a cada passo. Esse cálculo torna-se inviável na prática, onde a matriz de busca é obtida ao ser aproximada por uma constante. Para tal, as próprias imagens do banco de dados são apresentadas ao algoritmo de busca como novas entradas após manipuladas em forma, textura, rotação, translação e escala. Um filtro gaussiano é aplicado para suavizar a sobreposição de todos os elementos treinados conforme:

$$\frac{\partial r_i}{\partial p_j} = \sum_{k=1}^{s} w(\delta c_{j,k})[r_{i(p+\delta c_{j,k})} - r_{i(p)}], \tag{3.31}$$

onde w é o *kernel* gaussiano e $\frac{\partial r_i}{\partial p_j}$ é o elemento na posição i, j da matriz R.

Além disso, a aproximação por uma constante torna-se válida pois a mesma condição inicial

de busca é considerada: textura média na forma média obtidas no modelo combinado AAM.

O processo iterativo de buscas para minimização do erro ao representar uma imagem por seu parâmetro de aparência c é realizado em quatro passos:

- projeta-se o modelo obtido no treinamento combinado de forma e textura em uma nova imagem de entrada e calcula-se o erro $E_{(p)} = r_{(p)}^T r_{(p)}$
- computa-se δp necessário para minimizar $E_{(p)}$. A estimativa do vetor de aparência+pose $\hat{p}$ é dada por:

$$\hat{p} = p + k\delta p, \tag{3.32}$$

  onde $\delta p = -Rr_{(p)}$. Inicialmente $k = 1$.

- com o novo parâmetro $\hat{p} = (\hat{c}^T \quad \hat{t}^T)$ calcula-se um novo modelo da imagem de entrada $(\hat{g}_s)$ e verifica-se o novo erro. A nova estimativa é dada por:

$$\hat{g}_s = \bar{g} + P_g P_{cg} \hat{c}. \tag{3.33}$$

  caso ocorra diminuição do erro mantêm-se os parâmetros novos. Caso contrário, volta-se ao passo anterior e diminui-se o valor de k;

- repete-se o procedimento até que ocorra convergência ou que se alcance o número máximo de iterações.

Para uma maior precisão e melhor convergência, o algoritmo AAM utiliza multi-resolução de quatro camadas. Todo o processo de treinamento de modelo de forma, textura, aparência e busca é realizado nas escalas de 0,25 , 0,5 , 0,75 e 1 . Durante o processo de aquisição de dados em uma imagem de entrada é realizada a busca da menor para a maior escala. O resultado do modelo gerado pode ser aferido através da diferença entre a imagem de entrada e o modelo sintético gerado obtido na última escala.

Nesse ponto, pode-se observar duas características do algoritmo AAM que serão modificadas e exploradas, apresentando seus resultados no Capítulo 5:

- a convergência do algoritmo de busca é dependente do ponto inicial para aplicação do modelo, sendo proposto um conjunto de pontos definidos na região vizinha delimitada pelo centro da face detectado pelo algoritmo Viola Jones;

- o processo iterativo de mudança no parâmetro de aparência c definido na Equação 3.32 ocorre de forma linear, sendo proposto uma alteração de parâmetros de forma híbrida, adicionando um termo não linear.

Capítulo 4

Classificação

O vetor de aparência c gerado na fase de busca do algoritmo AAM representa uma das expressões faciais utilizadas na fase de treinamento e será empregado como entrada para o classificador, último bloco do sistema de reconhecimento de expressões faciais.

Como primeiro classificador foi utilizado o método dos k vizinhos mais próximos - k-NN e, em seguida, a máquina de vetores de suporte - SVM.

4.1 k-NN

O k-NN é um método não paramétrico para estimar uma densidade de probabilidade (DUDA; HART; STORK, 2001). A probabilidade P de um vetor x estar contido em uma região $\Re$ é:

$$P = \int_{\Re} p(x')dx'. \tag{4.1}$$

Se trabalharmos com um conjunto de amostras i.i.d. (independentes e igualmente distribuídos), a probabilidade de k dessas n amostras estarem contidas em $\Re$ é dada pela distribuição binomial segundo:

$$P_k = \binom{k}{n} P^k (1-P)^{n-k}. \tag{4.2}$$

Consequentemente, o valor esperado é dado por $\epsilon[k] = Pn$. Como na distribuição binomial para k há uma concentração forte em torno da média é possível escrever que $P = \frac{k}{n}$.

Se p_x for considerado contínuo e consideramos V como o volume delimitado por $\Re$:

$$\int_{\Re} p(x')dx' \cong p(x)V \cong \frac{k}{n}. \tag{4.3}$$

Com um número adequado e representativo de amostras a estimativa de $p(x)$ pode ser dada por

$$p(x) \cong \frac{k/n}{V}. \tag{4.4}$$

Para um número limitado de amostras temos:

- as regiões R_1, R_2, ..., R_n contém x;
- V_n é o volume definido por R_n;
- R_n contém k_n amostras.

Portanto, a n-ésima estimativa para $p(x)$ é expressa como $p(x) \cong \frac{k_n/n}{V_n}$. Para convergência devemos obter:

- $\lim_{x\to\infty} V_n = 0$;
- $\lim_{x\to\infty} k_n = \infty$;
- $\lim_{x\to\infty} \frac{k_n}{n} = 0$.

Portanto, a estimativa da densidade de probabilidade depende da escolha de V_n. No caso da Janela de Parzen especifica-se um volume V_n (DUDA; HART; STORK, 2001) e verifica-se quantas amostras estão contidas no espaço delimitado por V_n. Por exemplo, $V_n = \frac{1}{\sqrt{n}}$. No

entanto, se V_n for pequeno muitas áreas ficarão sem amostras envolvidas e a estimativa da densidade será igual a zero. Com grandes V_n haverá perda de resolução, pois regiões sem amostras recebem contagem de amostras muito distantes e não poderemos escrever a Equação 4.3. Nos dois casos, a densidade de probabilidade será errada. Uma alternativa é especificar V_n em função do banco de treino, fazendo V_n crescer em torno de um ponto até envolver k amostras. Ou seja, a densidade de probabilidade pode variar de acordo com o número k escolhido, conforme Figura 4.1.

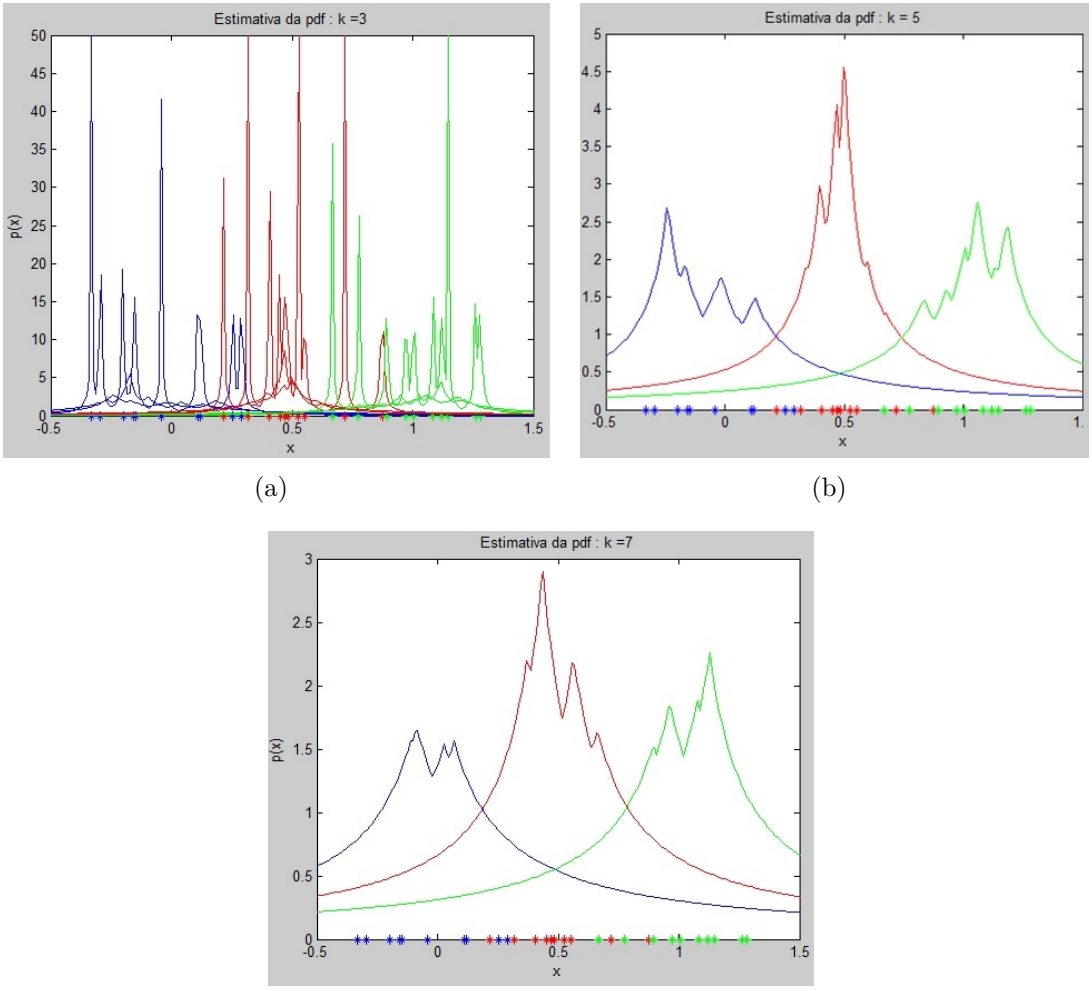


Figura 4.1: A escolha do parâmetro k altera a estimativa de densidade de probabilidade $p_{(x)}$. (a) k=3 (b) k=5 (c) k=7

Observa-se que a escolha adequada de k é fundamental para uma boa estimativa de densidade de probabilidade. À medida que diminui-se k, como na Figura 4.1(a), são formados picos de densidade concentrados em torno da amostra. Quando aumenta-se k, como na Figura 4.1(c), a região de influência de uma amostra aumenta e há um espalhamento da densidade de probabilidade.

Para a classificação de dados pode-se utilizar a densidade de probabilidade estimada para o conjunto de C classes, conforme se observa na expressão abaixo (COVER; HART, 1967)

$$P_n(\omega_i|x) = \frac{p_n(x,\omega_i)}{\sum_{j=1}^{C} p_n(x,\omega_j)} = \frac{k_i}{k}, \tag{4.5}$$

onde k_i é a quantidade de amostras da i-ésima classe ω_i das k amostras que caíram dentro do volume V empregado para estimar probabilidade. Observa-se, portanto, que com o k-NN é possível chegar a um classificador bayesiano, diferentemente da abordagem via Janela de Parzen.

Um dado de entrada é classificado como pertencente a uma das classes de treinamento maximizando a pdf, ou seja, é classificado como pertencente a classe que possui o maior número de amostras em seu entorno, conforme Figura 4.2. Desta forma, para o k-NN, uma vez escolhido a priori o valor de k, a probabilidade de ocorrência da classe é medida aplicando-se a Equação 4.5 que ajusta o volume V de modo a que sempre caiam k amostras no interior de V.

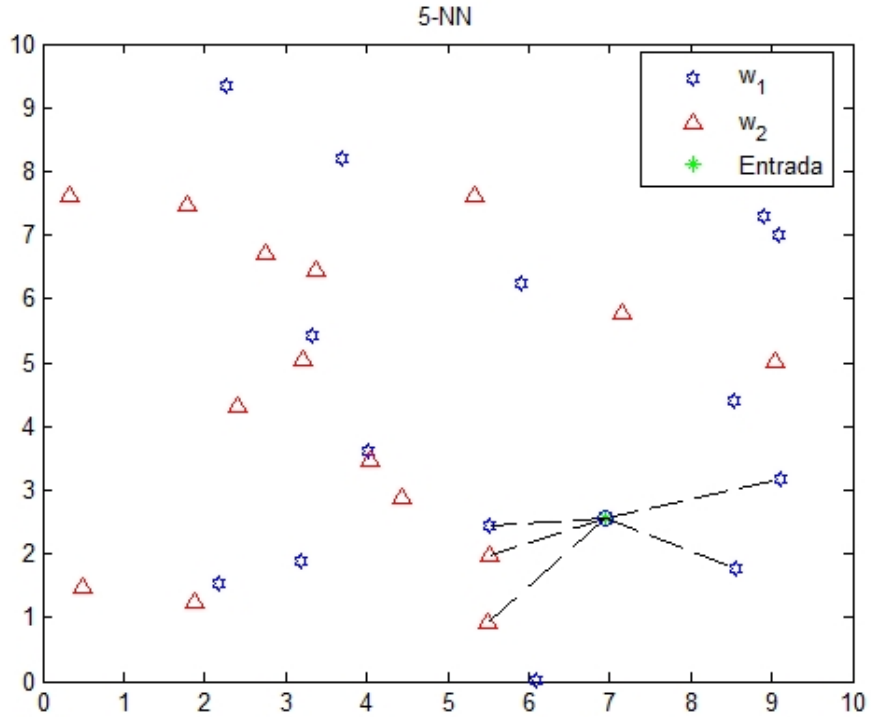


Figura 4.2: O dado de entrada circulado foi rotulado como pertencente à classe w_1 utilizando o 5-NN porque 3 dos vizinhos mais próximos são da classe w_1 contra 2 vizinhos da classe w_2.

A decisão de classe é feita na sequência de passos a seguir.

– para um protótipo de entrada X calcule a distância entre X e os n protótipos de treino $[X_1X_2X_3...X_n]$;

– verifique a classe dos k exemplos de treinos mais próximos do dado de entrada;

– atribua a entrada a classe com maior incidência, ou seja, a classe que apresenta o maior valor k_i/k.

Como o nome da técnica induz, para uma entrada x é necessário contabilizar os k vizinhos mais próximos de x. Para isto, é necessário utilizar algum método de medir distância. Nesse trabalho foi utilizada a Distância Euclidiana. Já o número de vizinhos k foi escolhido utilizando a validação cruzada, onde o 3-NN apresentou a menor taxa de erro. Se considerarmos o caso do 1-NN, o vizinho mais próximo, o resultado do classificador pode ser visto como um Diagrama de Voronoi, ilustrado na 4.3, onde cada região possui uma demarcação para associação de classes. Nesse caso bimodal as classes são círculos (em vermelho) e hexágonos (em verde).

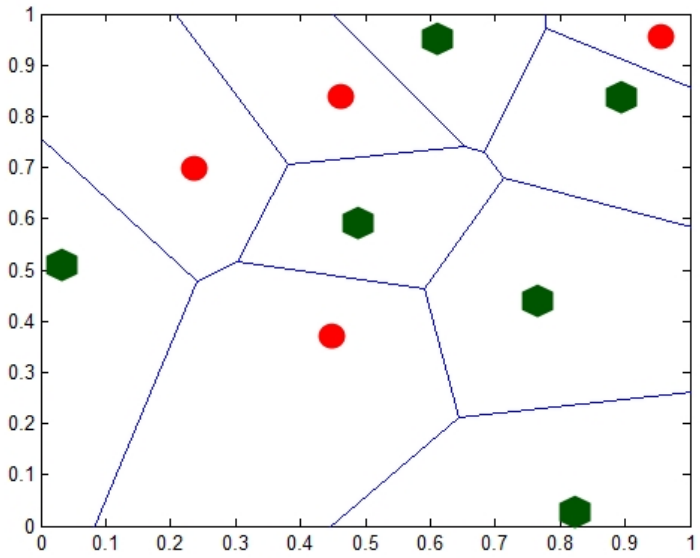


Figura 4.3: As regiões demarcadas no Diagrama de Voronoi como pertencentes a 2 classes distintas.

O erro obtido com o k-NN, no caso de 1-NN, é no máximo duas vezes o erro obtido se houvesse a real distribuição dos dados, estimados segundo a regra de Bayes (DUDA; HART; STORK, 2001). Portanto, temos uma boa estimativa do desempenho do classificador.

4.2 SVM

O SVM - *Support Vector Machine* , ou máquina de vetores de suporte, é um método de aprendizagem supervisionada utilizado para classificação de dados (CORTES; VAPNIK, 1995). Em oposição ao k-NN, é um método não probabilístico.

Primeiramente, é realizada uma fase de treinamento onde os dados do banco de treino são apresentados ao algoritmo. Cada uma das x_i entradas de treinamento, onde i é o índice da entrada, possuem um alvo associado y_i que indica a qual classe pertence o dado. O SVM deve ser capaz de rotular um novo dado de entrada atribuindo um valor para seu alvo. Portanto, o aprendizado é realizado de maneira indutiva.

O foco do SVM é conseguir separar os dados que não são linearmente separáveis utilizando, para isso, uma transformação não linear onde os dados são levados para um espaço de maior dimensão denominado espaço de características através de uma função *kernel*. Nesse novo espaço deve existir um ou mais hiperplanos capazes de separar os dados. Os hiperplanos no espaço de maior dimensão são definidos como o conjunto de pontos cujo produto escalar dos vetores é constante. O objetivo pode ser classificar um novo dado de entrada não utilizado no treino em uma das possíveis classes treinadas ou realizar uma regressão.

4.2.1 Duas Classes Linearmente Separáveis

Iniciaremos o estudo do SVM apresentando o caso de um banco de dados divididos em duas classes ω_1 e ω_2 linearmente separáveis. Cada elemento do banco é bidimensional e, portanto, apresenta duas componentes de características.

A Figura 4.4 ilustra as classes ω_1 e ω_2 composta por elementos que possuem distribuição normal com desvio padrão de 0,5 e 1,5 para suas características em uma distribuição Normal. A classe ω_1 possui média amostral igual a zero ao passo que ω_2 possui média 3. Observa-se que com uma reta é possível separar as classes. No entanto, qual deve ser essa reta, que em maior dimensões pode ser vista como um hiperplano, já que existem várias possíveis capazes de executar a tarefa como, por exemplo, as retas r_1, r_2, r_3 e r_4? Deve existir um compromisso entre a capacidade de separar as classes do banco e a capacidade de generalização para uma correta classificação de novas entradas.

A superfície de decisão $g(\mathbf{x})$ de um espaço l-dimensional para um grupo de dados com l

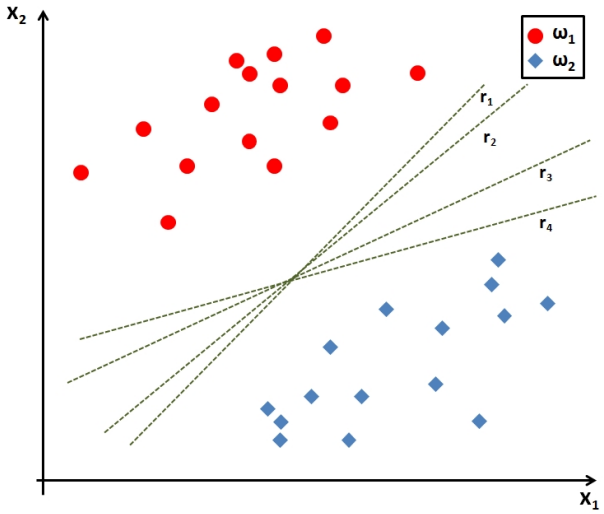


Figura 4.4: As classes ω_1 e ω_2 são linearmente separáveis. As retas r_1, r_2, r_3 e r_4 são exemplos de limites de decisão.

características é definido pela equação:

$$g(x) = \mathbf{w}^T\mathbf{x} + w_0 = 0, \tag{4.6}$$

onde $\mathbf{w} = [w_1\ w_2\ w_3\ ...\ w_l]^T$ é denominado vetor de pesos e w_0 é o limiar de *offset*. Quando $w_0 = 0$, o hiperplano de decisão corta a origem. Se consideramos dois pontos $\mathbf{x}_1$ e $\mathbf{x}_2$ contidos no hiperplano de decisão, verificamos

$$\mathbf{w}^T\mathbf{x}_1 = 0 = \mathbf{w}^T\mathbf{x}_2 \tag{4.7}$$

$$\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_2) = 0 \tag{4.8}$$

$$\mathbf{w}^T\mathbf{x}_3 = 0. \tag{4.9}$$

Como o resultado $\mathbf{x}_3 = \mathbf{x}_1 - \mathbf{x}_2$ é um ponto que também pertence ao hiperplano separador, pode-se concluir que o $\mathbf{w}$ é ortogonal ao hiperplano de decisão, uma vez que $\mathbf{w}^T\mathbf{x}_3 = 0$.

Observando a Figura 4.5, verifica-se que para um determinado vetor de peso $\mathbf{w} = [w_1\ w_2]$ é possível determinar a distância z entre uma amostra x e o hiperplano fazendo

$$z = \frac{|g(\mathbf{x})|}{\sqrt{w_1^2 + w_2^2}} = \frac{|g(\mathbf{x})|}{\|\mathbf{w}\|}, \tag{4.10}$$

e a distância d entre a origem e o hiperplano na mesma orientação de $\mathbf{w}$ segundo

$$d = \frac{|w_0|}{\sqrt{w_1^2 + w_2^2}} = \frac{|w_0|}{\|\mathbf{w}\|}. \tag{4.11}$$

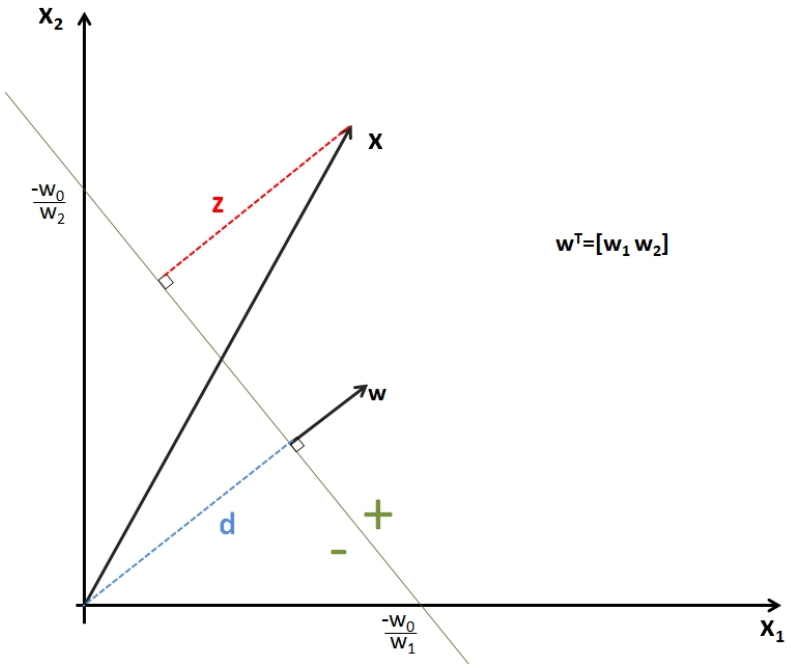


Figura 4.5: Medidas entre uma amostra $\mathbf{x}$ e o hiperplano separador.

A distância z é a menor distância entre uma amostra de determinada classe e o hiperplano separador. Observe na Figura 4.6 que a escolha de um z está diretamente relacionada à relação de compromisso entre separar corretamente as classes de treino e na capacidade de generalização para novas entradas.

A margem M, conforme Figura 4.6, é a distância que separa as classes segundo escolha de $\mathbf{w}$. É desejável a maximização da margem para as duas classes, portanto é comum o hiperplano ser definido de modo a estar na metade da distância entre as amostras mais próximas do hiperplano para as duas classes.

Portanto, $\mathbf{w}$ deve ser escolhido de forma que as seguintes relações sejam satisfeitas para o caso de duas classes:

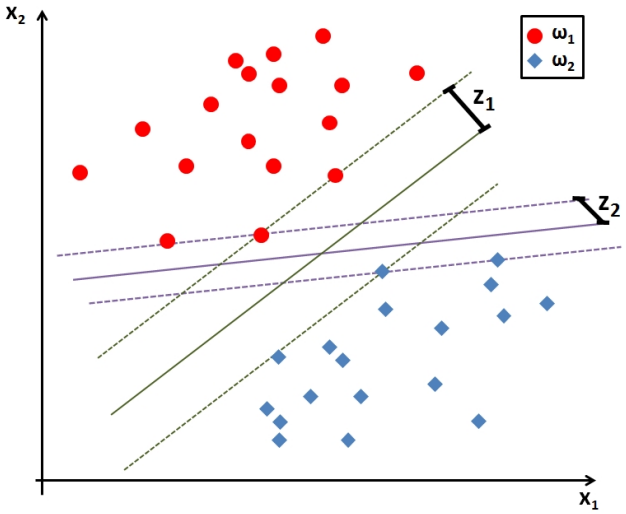


Figura 4.6: A escolha de $\mathbf{w}$ e da margem está relacionada com a capacidade de separar os dados de treino e na generalização do classificador para novos dados de entrada.

$$\mathbf{w}^T x > 0, \qquad \forall x \in \omega_1 \tag{4.12}$$

$$\mathbf{w}^T x < 0, \qquad \forall x \in \omega_2, \tag{4.13}$$

como ocorre no exemplo da Figura 4.6. Os dados à direita do hiperplano separador possuem saídas positivas ao passo que os dados à esquerda do hiperplano separador possuem saídas negativas. Esse é o típico caso onde deve ser escolhida uma função de custo apropriada e um posterior algoritmo de otimização. Um caso típico é do *perceptron*, onde através da descida de gradiente a função de custo $J(\mathbf{w}) = \sum_{\mathbf{x} \in Y} \sigma_x \mathbf{w}^T \mathbf{x}$ é minimizada, onde Y é o conjunto dos dados de treino e σ_x é a função capaz de separar as classes conforme Equações 4.12 e 4.13.

A Figura 4.7 apresenta a margem M que melhor separa linearmente as classes.

Para o SVM, iniciamos procurando por $\mathbf{w}$ que maximize a margem. Realiza-se um escalamento de $\mathbf{w}$ e w_0 de maneira que $g(\mathbf{x}) = +1$ para o ponto mais próximo pertencente a ω_1 e $g(\mathbf{x}) = -1$ para o ponto mais próximo de ω_2.

Formalmente, a margem M é definida como a distância entre dois hiperplanos paralelos que satisfazem $\mathbf{w}^T \mathbf{x} + w_0 = \pm 1$.

Isso é equivalente a definir a margem M como

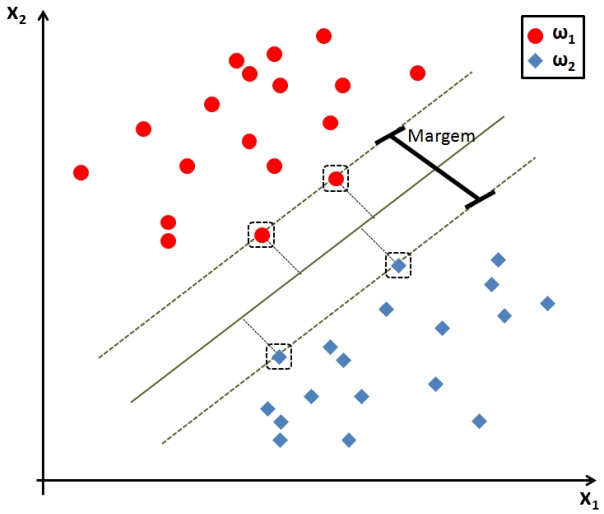


Figura 4.7: Hiperplano que apresenta melhor margem.

$$M = \frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \tag{4.14}$$

obedecendo às restrições

$$\mathbf{w}^T\mathbf{x} + w_0 > +1, \qquad \forall\ \mathbf{x} \in \omega_1 \tag{4.15}$$

$$\mathbf{w}^T\mathbf{x} + w_0 < -1, \qquad \forall\ \mathbf{x} \in \omega_2. \tag{4.16}$$

A solução que maximiza a margem M é a máquina de suporte de vetores linear ou LSVM.

Os parâmetros $\mathbf{w}$ e w_0 do hiperplano são computados com as seguintes ações:

- minimiza-se a função de custo $J \equiv \frac{1}{2}\|\mathbf{w}\|^2$, conhecido como problema primal, e
- sujeito a $y_i(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1, \quad i = 1, 2, ..., N,$

onde i é o índice para as N amostras e $y_i = +1$ é o alvo para a amostra x_i se $x_i \in \omega_1$ e $y_i = -1$ se $x_i \in \omega_2$. Observe que a função de decisão de classe é $sinal(\mathbf{w}x^T + w_0)$ e que a minimização da norma maximiza a margem.

Essa otimização não linear do tipo quadrática com uma inigualdade como restrição possui

solução que deve seguir as condições de Karush-Kuhn-Tucker - KKT (BISHOP, 2006), conforme as Equações a 4.17 a 4.20.

$$\frac{\partial}{\partial \mathbf{w}}\mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda}) = \mathbf{0} \tag{4.17}$$

$$\frac{\partial}{\partial w_0}\mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda}) = \mathbf{0} \tag{4.18}$$

$$\lambda_i \geq 0, \quad i = 1, 2, .., N \tag{4.19}$$

$$\lambda_i[y_i(\mathbf{w}^T\mathbf{x}_i - 1)] = 0, \quad i = 1, 2, .., N, \tag{4.20}$$

onde $\boldsymbol{\lambda}$ é o vetor multiplicador de Lagrange, λ_i, para $i = 1, 2, .., N$, são os multiplicadores de Lagrange e $\mathcal{L}$ é a Função de Lagrange ou lagrangiano, definido como:

$$\mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \lambda_i[y_i(\mathbf{w}^T\boldsymbol{x_i} + w_0) - 1] = 0. \tag{4.21}$$

Combinando a Equação 4.21 com as restrições de 4.17 e 4.18, temos:

$$\mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x_i} \quad e \tag{4.22}$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0. \tag{4.23}$$

Como os multiplicadores de Lagrange λ_i são iguais a zero ou valores positivos, $\lambda_i \geq 0$, a solução ótima para $\mathbf{w}$ será composta pela combinação linear dos N_s vetores de suporte associados aos $\lambda_i \neq 0$, onde $N_s \leq N$. Por definição, os vetores de suporte são obtidos como

$$\sum_{i=1}^{N_s} \lambda_i y_i = 0. \tag{4.24}$$

Devido à restrição imposta na Equação 4.20, o vetor de suporte está contido na solução de $\mathbf{w}^T\mathbf{x} + w_0 = \pm 1$. Em outras palavras, os vetores de suporte são escolhidos como os elementos mais críticos dos dados de treinamento do ponto de vista do classificador linear.

O parâmetro w_0 é obtido considerando a média dos valores de $\lambda_i \neq 0$ que satisfazem a Equação 4.20.

É importante observar que a formulação das restrições e o fato de que a função de custo é estritamente convexa garantem um hiperplano ótimo separador único.

Por fim, a maximização do Lagrangiano é obtida resolvendo o problema de otimização (THEODORIDIS; KOUTROUMBAS, 2009), conhecido como problema dual

$$\max_{\boldsymbol{\lambda}} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{x_j} \right) \tag{4.25}$$

sujeito às restrições

$$\sum_{i=1}^{N} \lambda_i y_i = 0 \tag{4.26}$$

e

$$\boldsymbol{\lambda} \geq 0. \tag{4.27}$$

4.2.2 Duas Classes Não Linearmente Separáveis

Os dados de entrada podem ser das mais variadas formas e, portanto, esperam-se casos onde há uma maior dificuldade de resolver o problema da classificação. É o caso de dados de classes distintas ω_1 e ω_2 que não são separáveis linearmente, ou seja, não existe um hiperplano capaz de separar todas as amostras pertencentes a ω_1 das amostras pertencentes a ω_2 do banco de treino. Esse é o caso das expressões faciais, já que existem componentes de características como, por exemplo, olhos em que verifica-se uma maior abertura de olhos para as expressões de surpresa e medo. Portanto, espera-se que parte das características que compõem a expressão facial não seja linearmente separável. A Figura 4.8 ilustra essa condição.

Nesse caso, definido qualquer hiperplano separador e uma dada margem M verifica-se que

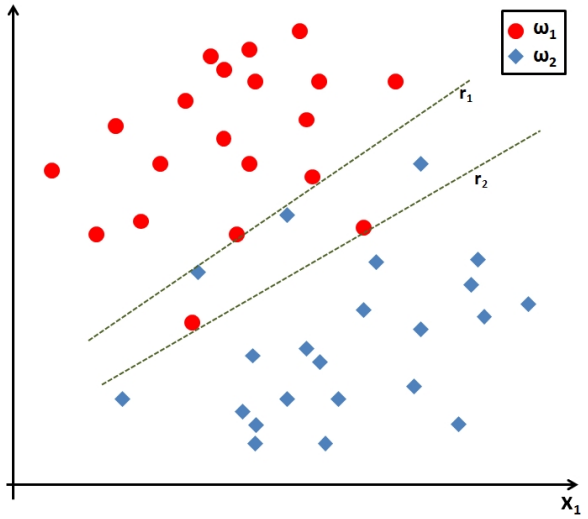


Figura 4.8: Duas classes não separáveis linearmente, isto é, não existe um hiperplano capaz de separar as amostras de ω_1 das amostras pertencentes a ω_2.

existem três possíveis situações:

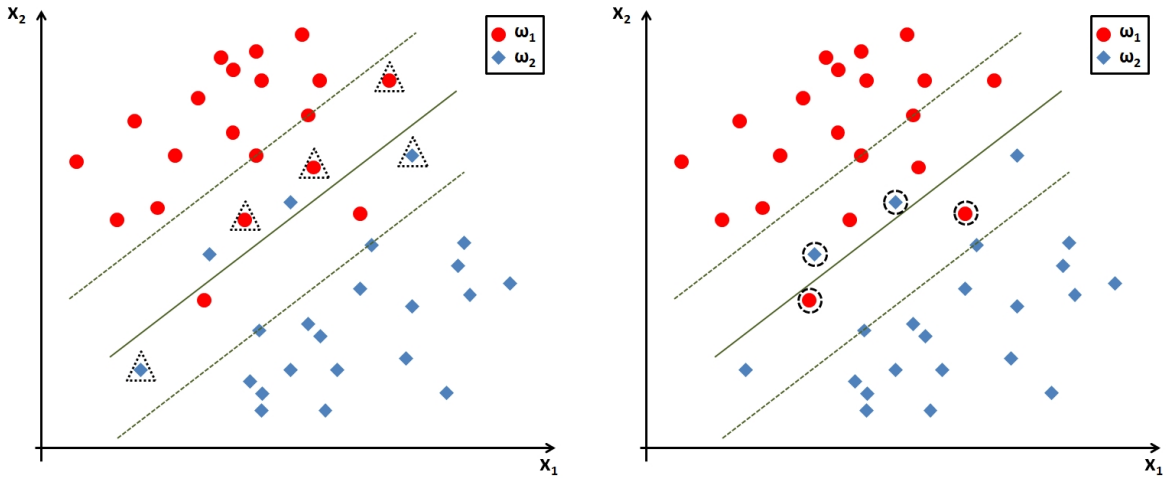
1. dados que são corretamente classificados, ficando fora dos limites definidos pela margem M;
2. dados que são corretamente classificados e que não respeitam a margem, conforme Figura 4.9(a);
3. e, por fim, dados que são erroneamente classificados, conforme Figura 4.9(b).

Para o primeiro caso, $\boldsymbol{\omega}^T\mathbf{x} + w_0 = \pm 1$, como no caso linear. Os dados corretamente classificados dentro da margem satisfazem $0 \leq y_i(\boldsymbol{\omega}^T\mathbf{x} + w_0) < 1$. Já os dados erroneamente classificados dentro da margem resultam em $y_i(\boldsymbol{\omega}^T\mathbf{x} + w_0) < 0$.

Os três casos podem ser generalizados introduzindo uma variável de folga ξ_i para a restrição conforme Equação 4.28.

$$y_i(\boldsymbol{\omega}^T\mathbf{x} + w_0) \geq 1 - \xi_i \tag{4.28}$$

O primeiro caso corresponde a $\xi_i = 0$, o segundo caso a $0 < \xi_i \leq 1$ e o último caso a $\xi_i > 1$. Portanto, alguns ajustes devem ser feitos na máquina de vetores de suporte para que seja satisfeita a relação de compromisso entre separar os dados de treino da melhor maneira



(a) (b)

Figura 4.9: Os dados de ω_1 e ω_2 que estão fora da faixa delimitada pela margem são corretamente classificados. Em (a), os dados envolvidos por triângulos são corretamente classificados, apesar de não respeitarem a margem. Já em (b), os dados destacados estão fora da margem e foram erroneamente classificados.

possível sendo capaz de manter a generalização para classificar corretamente novos dados de entrada. Isso equivale a manter a maior margem possível que mantenha o menor número de pontos possíveis com $\xi_i > 0$. A função de custo é modificada levando-se em conta a variável de folga ξ_i:

$$J(\mathbf{w}, w_0, \boldsymbol{\xi}) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} I(\xi_i), \tag{4.29}$$

onde $\boldsymbol{\xi}$ é o vetor contendo os parâmetros ξ_i e

$$I(\xi_i) = \begin{cases} 1, & \text{se } \xi_i > 0 \\ 0, & \text{se } \xi_i = 0. \end{cases} \tag{4.30}$$

O parâmetro C é uma variável que regula o peso a ser considerado para cada um dos dois termos concorrentes da função de custo.

Como a função $I(.)$ é descontínua, a função de custo é modificada e o hiperplano será escolhido minimizando a função de custo

$$J(\mathbf{w}, w_0, \boldsymbol{\xi}) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \tag{4.31}$$

sujeito às restrições

$$y_i(\boldsymbol{\omega}^T\mathbf{x} + w_0) \geq 1 - \xi_i \tag{4.32}$$

$$\xi_i \geq 0, \quad i = 1, 2, ..., N. \tag{4.33}$$

Para essas condições, o Lagrangiano correspondente é

$$\mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i=1}^{N}\lambda_i[y_i(\mathbf{w}^T\boldsymbol{x_i} + w_0) - 1] = 0. \tag{4.34}$$

As condições KKT são:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = \mathbf{0} \quad \rightarrow \quad \mathbf{w} = \sum_{i=1}^{N}\lambda_i y_i x_i \tag{4.35}$$

$$\frac{\partial\mathcal{L}}{\partial w_0} = 0 \quad \rightarrow \quad \mathbf{w} = \sum_{i=1}^{N}\lambda_i y_i \tag{4.36}$$

$$i = 1, 2, ..., N \tag{4.37}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = \mathbf{0} \rightarrow C - \mu_i - \lambda_i = 0 \tag{4.38}$$

$$\lambda_i[y_i(\mathbf{w}^T\boldsymbol{x_i} + w_0) - 1 + \xi_i] = 0 \tag{4.39}$$

$$\mu_i\xi_i = 0 \tag{4.40}$$

$$\xi_i \geq 0 \tag{4.41}$$

$$\mu_i \geq 0. \tag{4.42}$$

$$\lambda_i \geq 0. \tag{4.43}$$

As condições combinadas correspondem a:

$$maximizar \quad \mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\mu}) \tag{4.44}$$

sujeito às restrições:

$$\mathbf{w} = \sum_{N}^{i=1} \lambda_i y_i x_i \tag{4.45}$$

$$\sum_{N}^{i=1} \lambda_i y_i = 0 \tag{4.46}$$

$$C - \mu_i - \lambda_i = 0 \tag{4.47}$$

$$\xi_i \geq 0 \tag{4.48}$$

$$\mu_i \geq 0. \tag{4.49}$$

$$i = 1, 2, ..., N. \tag{4.50}$$

Substituindo as restrições no Lagrangiano, obtemos como solução para o problema de otimização (THEODORIDIS; KOUTROUMBAS, 2010)

$$\max_{\boldsymbol{\lambda}} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{x_j} \right) \tag{4.51}$$

com as restrições

$$0 \leq \lambda_i \leq C, \quad i = 1, 2, ..., N \tag{4.52}$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0 \tag{4.53}$$

Portanto, a Máquina de Vetores de Suporte pode ser utilizada na classificação de dados com duas classes que não são linearmente separáveis desde que seja apresentada uma condição relaxação através da variável de folga. A maximização da Margem visa generalizar o aprendizado e classificar com menor erro possível novos dados.

4.2.3 Caso Multiclasses

Até o momento foi considerado apenas a classificação envolvendo duas classes. Para o caso de T-classes pode-se realizar o problema dividindo-o em T problemas de duas classes, onde a cada divisão é considerado o subproblema de distinguir uma classe específica de todas as outras. Portando, trata-se da abordagem um contra todos (*one against all*) (THEODORIDIS; KOUTROUMBAS, 2009). O caso de três classes ω_1, ω_2 e ω_3 é ilustrado na Figura 4.10. Verifica-se que o hiperplano delimitado pela reta r_1 é capaz de separar ω_1 de ω_2 e ω_3. Já r_2 separa ω_2 de ω_1 e ω_3. Por fim, r_3 é o hiperplano separador de ω_3 em relação a ω_1 e ω_2.

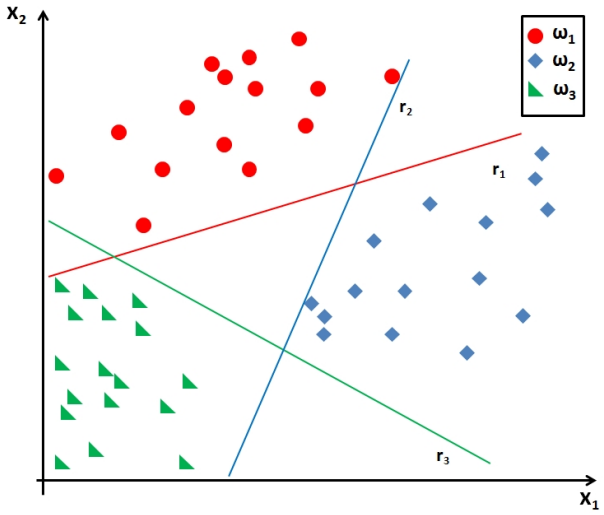


Figura 4.10: .O problema de classificação multiclasse com T=3 pode ser resolvido através de 3 problemas de duas classes do tipo uma classe contra todas. O hiperplano r_1 separa ω_1 de ω_2 e ω_3. Já r_2 e r_3 isolam as classes ω_2 e ω_3, respectivamente.

Nesse caso, devemos ter para cada classe uma função discriminante $g_i(\mathbf{x})$, $i = 1, 2, ..., T$ que, para a classe ω_i, em relação a todas as outras $T-1$ classes, satisfaça:

$$g_i(\mathbf{x}) > g_j(\mathbf{x}), \quad \forall j \neq i, \quad se\ \mathbf{x} \in \omega_i. \tag{4.54}$$

Utilizando o SVM na função discriminante, deve-se obter $g_i(\mathbf{x}) = 0$ como o hiperplano ótimo capaz de separar ω_i de todas as classes . O resultado dessa abordagem fornece $g_i(\mathbf{x}) > 0$ se $\mathbf{x} \in \omega_i$ e $g_i(\mathbf{x}) < 0$ caso contrário. Finalmente, a classificação de uma entrada $\mathbf{x}$ desconhecida é realizada fazendo

$$\mathbf{x} \in \omega_i \quad se \quad i = arg \max_t [g_t(\mathbf{x})], \tag{4.55}$$

onde $t = 1, 2, ..., T$.

Uma desvantagem dessa abordagem é que existem regiões do espaço que não são definidas como pertencentes a nenhuma das classes especificamente, conforme Figura 4.11. A região Ω pode pertencer a qualquer uma das classes ω_1, ω_3 ou ω_3, dependendo de quais hiperplanos são considerados. Uma outra consideração deve ser ponderada na etapa de treinamento em relação ao número de amostras. Como é um problema de uma classe contra T-1, é possível que a quantidade de exemplos negativos sempre superem os positivos. Isso acarreta a necessidade de possuir um banco de treino com o maior número de amostras possíveis.

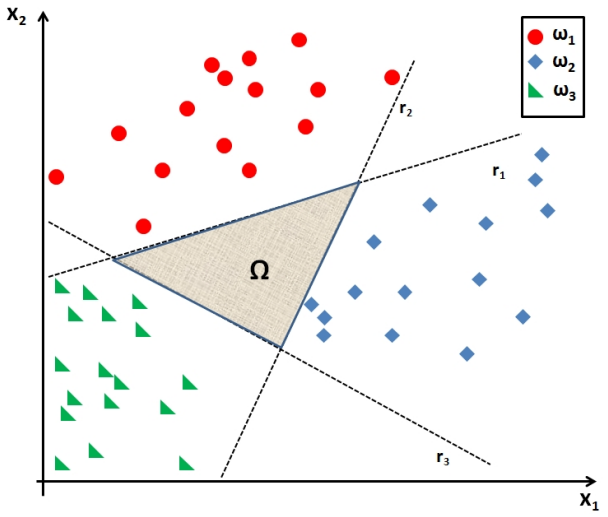
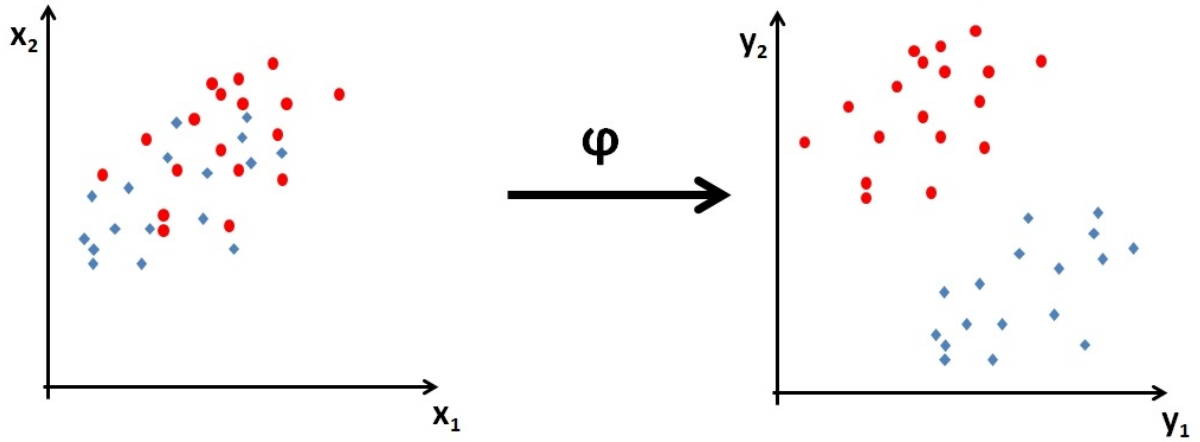


Figura 4.11: Os hiperplanos classificadores delimitadores definidos por r_1, r_2 e r_3 deixam uma região do espaço definida por Ω como indefinida para o problema de classificação.

4.2.4 Utilização de *Kernels*

Apesar de ser possível o cálculo de vetores de suporte para classes que não são linearmente separáveis, existem distribuições, como ilustrado na Figura 4.12(a). A natureza dos dados resulta em hiperplanos que geram muitos erros de classificação. Nesse caso, há uma sobreposição da distribuição entre as classes.

Uma alternativa para esse tipo de dados é gerar um mapeamento onde os dados são levados para um novo espaço, denominado espaço de características, para que sejam linearmente separáveis. A Figura 4.12(b) apresenta uma transformação φ capaz de representar os dados originais em um espaço de características onde é possível separar os dados através de um hiperplano. Ou seja, nesse novo espaço é viável a utilização da máquina de vetores de suporte.



(a) (b)

Figura 4.12: (a)As classes ω_1 e ω_2 apresentam distribuições com superposição dos dados, acarretando erros na classificação através do SVM.(b)Os dados originais são remapeados através da transformação φ , sendo linearmente separáveis no novo espaço que apresenta dimensão maior que o espaço original.

Considere um conjunto de N dados onde cada padrão ou protótipo $\mathbf{x}_i$ apresenta d características, isto é, $\mathbf{x} \in \mathcal{X} \subseteq \Re^d$ e $i = [1, 2, ...N]$. As informações explícitas em cada protótipo podem ser inspecionadas analisando as características que compõem os dados, ou seja, $[x_{i1}\ x_{i2}\ ...\ x_{id}]$. Adicionalmente, pode-se extrair atributos implícitos através de uma transformação $\Phi(\mathbf{x})$ que mapeia os dados originais em um espaço de características de maior dimensão. Espera-se que nesse novo espaço os dados sejam linearmente separáveis. Uma opção para a transformação $\Phi(\mathbf{x})$ é utilizar os resultados dos produtos das d características.

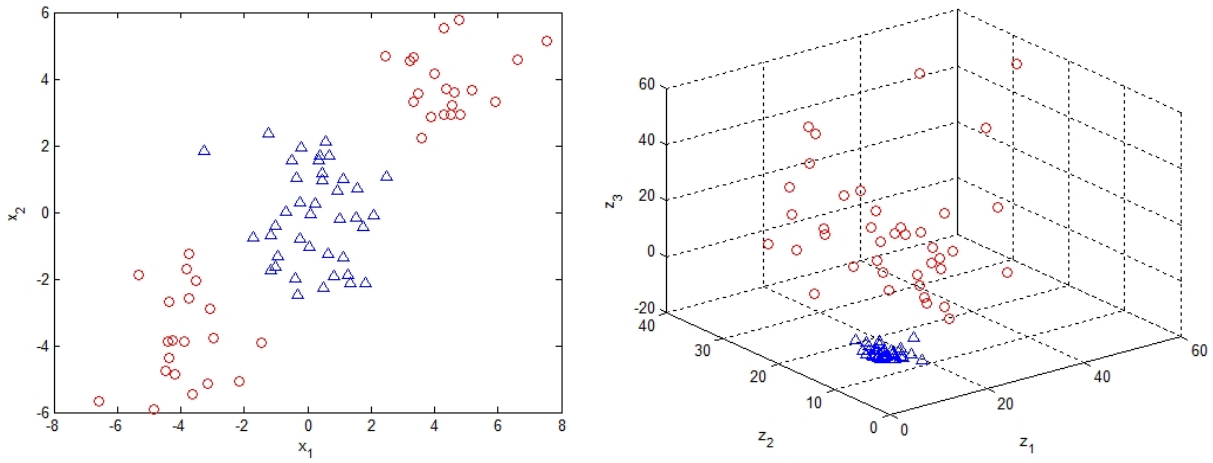
Por exemplo, considere dados em um espaço $\Re^2$ transformados por $\Phi(\mathbf{x})$ em um espaço de 3 dimensões, conforme

$$\mathbf{x} = [x_1\ x_2] \quad \in \quad \Re^2 \quad \overset{\Phi(\mathbf{x})}{\longmapsto} \quad \mathbf{z} = [z_1\ z_2\ z_3] \quad \in \quad \Re^3, \tag{4.56}$$

utilizando a relação

$$[x_1\ x_2] \quad \longmapsto \quad [x_1^2\ x_2^2\ \sqrt{2}x_1x_2]. \tag{4.57}$$

Observe nas Figuras 4.13(a) e 4.13(b) o resultado desse remapeamento e a possibilidade de um hiperplano separador no novo espaço capaz de separar linearmente as duas classes.



(a) (b)

Figura 4.13: Dados não linearmente separáveis no espaço $\Re^2$ em (a) são separáveis no espaço de características $\Re^3$ em (b) através de um remapeamento $\Phi(\mathbf{x})$.

Portanto, o mapeamento pode ser realizado utilizando todos os monômios das características originais. Esse procedimento pode se tornar custoso computacionalmente quando aumenta-se o número de características d, pois são possíveis $\frac{(d+N-1)!}{N!(d-1)!}$ monômios. Para o caso de imagens, o número de monômios referentes a cada pixel torna o mapeamento impraticável.

Uma alternativa para representar os dados em um novo espaço é realizar o mapeamento de forma implícita, isto é, sem conhecer a função $\Phi(\mathbf{x})$, mas sabendo a relação entre os dados no novo espaço. Observe que no exemplo anterior o resultado $\Phi(\mathbf{x})\Phi(\mathbf{x})$ pode ser calculado através do produto interno

$$\begin{aligned} \mathbf{x} \cdot \mathbf{x} &= \langle \mathbf{x}, \mathbf{x} \rangle && (4.58)\\ &= \mathbf{x}\mathbf{x}^T && (4.59)\\ &= [x_1^2 \; x_2^2 \; \sqrt{2}x_1x_2][x_1^2 \; x_2^2 \; \sqrt{2}x_1x_2]^T && (4.60)\\ &= x_1^2 + x_2^2 + 2x_1x_2. && (4.61) \end{aligned}$$

Formalmente, a transformação é realizada através de uma função *kernel* definida como $K(\boldsymbol{x_i}, \boldsymbol{x_j})$, definida pela Equação 4.62.

$$K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j). \quad (4.62)$$

Observe que a utilização de um *kernel* motiva o problema de classificação, uma vez que a distância no espaço de características, ainda que a transformação $\Phi(\mathbf{x})$ não seja conhecida é possível através do próprio *kernel* com sua função explícita. Para o exemplo dado, a distância entre dois dados no espaço de características $D(\mathbf{z}_1, \mathbf{z}_2)$ é obtida com as relações:

$$\begin{aligned} D(\mathbf{z}_1, \mathbf{z}_2) &= |\mathbf{z}_1 - \mathbf{z}_2| && (4.63)\\ &= \sqrt{(\mathbf{z}_1 - \mathbf{z}_2) - (\mathbf{z}_1 - \mathbf{z}_2)} && (4.64)\\ &= \sqrt{(\mathbf{z}_1 \cdot \mathbf{z}_1) - (2\mathbf{z}_1 \cdot \mathbf{z}_2)(\mathbf{z}_2 \cdot \mathbf{z}_2)} && (4.65)\\ &= \sqrt{K(\mathbf{x}_1, \mathbf{x}_1) - 2K(\mathbf{x}_1, \mathbf{x}_2) + K(\mathbf{x}_2, \mathbf{x}_2)} && (4.66) \end{aligned}$$

Porém encontrar *kernels* não é uma tarefa trivial, pois deve-se definir explicitamente um espaço vetorial $\mathcal{H}$, associado com uma transformação $\Phi : \mathcal{X} \to \mathcal{H}$, tal que em $\mathcal{H}$ os padrões mapeados sejam linearmente separáveis e o *kernel* é calculado via um produto interno, dependendo exclusivamente dos padrões de entrada e ignorando por completo a forma explícita de Φ.

De fato, adota-se o caminho inverso. Ou seja, partindo-se de um *kernel*, define-se $\mathcal{H}$ e Φ de forma que a relação $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ seja satisfeita e o cálculo viabilizado computacionalmente através de um produto escalar. A justificativa matemática está contida no Teorema de Mercer (COURANT; HILBERT, 1953) que estabelece o conceito para o *Kernel* Postivo Definido: uma função $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ tal que

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \tag{4.67}$$

onde $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in \mathcal{X}$ e $\beta_1, ..., \beta_n \in \mathbb{R}$ para todo $n \in \mathcal{N}$.

Em seguida, utiliza-se o *Kernel* Reduzido no Espaço de Hilbert (*Reduced Kernel Hilbert Space* - RKHS). Seja $K : \mathcal{X} \times \mathcal{X} \to$ um *kernel* positivo definido, a função $\mathcal{X} \ni \mathbf{x} \mapsto K(\mathbf{x}, \bullet)$ é denominado *kernel* parcialmente avaliado. O mesmo ocorre para a função $\mathcal{X} \ni \mathbf{x} \mapsto K(\bullet, \mathbf{x})$. Os *kernels* positivos avaliados apresentam as seguintes propriedades (HOFMANN; SCH; SMOLA, 2008).

1. $K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_i, \bullet) \cdot K(\bullet, \mathbf{x}_j)$;

Kernel	$\mathbf{K}(\boldsymbol{x_i}, \boldsymbol{x_j})$
Linear	$\boldsymbol{x_i}^T \boldsymbol{x_j}$
Polinomial	$(\gamma \boldsymbol{x_i}^T \boldsymbol{x_j})^d$, $\gamma > 0$
Polinomial Não Homogêneo	$(\gamma \boldsymbol{x_i}^T \boldsymbol{x_j} + r)^d$, $\gamma > 0$ $r > 0$
RBF	$e^{(-\gamma \|\boldsymbol{x_i} - \boldsymbol{x_j}\|)}$
Sigmoidal	$tanh(\gamma \boldsymbol{x_i}^T \boldsymbol{x_j} + r)$

Tabela 4.1: Principais *kernels* utilizados para classificação de dados não linearmente separáveis através da máquina de vetores de suporte: linear, polinomial, *radial basis function* e Sigmoidal.

2. $K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_i)$.

Por fim, verifica-se que $\mathcal{H}$ é um espaço vetorial com produto interno e $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$.

Para o exemplo inicial $\Re^2 \to \Re^3$, temos

$$\Phi(x_i) \cdot \Phi(x_j) = (\mathbf{x} \cdots \bullet)^2 \cdots (\bullet \cdots \mathbf{x})^2 = K(\mathbf{x}_i, \mathbf{x}_j). \tag{4.68}$$

A Tabela 4.1 apresenta os *kernels* mais utilizados na literatura.

Observe que as Equações 4.25 e 4.51 representam a forma dual do problema de otimização e apresentam em seus cálculos um termo em função produto interno, motivando a implementação do *kernel*.

Portanto, os dados não separáveis linearmente, podem ser classificados através da utilização de uma máquina de vetores de suporte e uma função *kernel* resolvendo o problema de otimização (HSU; CHANG; LIN, 2010)

$$\min_{\substack{\mathbf{w} \\ b_0 \\ \boldsymbol{\xi}}} \left(\frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{N} \xi_i \right) \tag{4.69}$$

sujeito às restrições

$$y_i(\mathbf{w}^T \phi(\boldsymbol{x_i}) + b_0) \geq 1 - \xi_i e \tag{4.70}$$

$$\xi_i \geq 0. \tag{4.71}$$

Capítulo 5

Resultados

Nesse trabalho é realizada a localização de faces utilizando o *framework* Viola-Jones baseado no *Adaboost*, a extração de características aplicando o modelo estatístico de aparência ativa AAM e a classificação empregando os classificadores k-NN e o SVM com *kernel* RBF. O fluxo de informação e tarefas de cada etapa é ilustrado na Figura 5.1.

Na etapa de treinamento ou etapa *offline*, um banco de dados com sete expressões faciais é utilizado. Para cada imagem desse banco é marcado um conjunto fixo de pontos, formando um conjunto de pontos de referência para a criação de um modelo estatístico AAM onde será avaliada a forma e textura de cada expressão facial baseado na disposição geométrica dos pontos aferidos e nas intensidades dos valores dos *pixels*. Em seguida, para cada imagem de treino é gerado um vetor de características ou vetor de aparência responsável pela parametrização do formato e textura da expressão facial baseado no modelo AAM. Associado à cada imagem de entrada há um vetor de alvo indicando a qual das classes de expressão facial a imagem pertence. Os vetores de características e seus alvos são entradas para a etapa de treinamento do classificador, onde realiza-se o cálculo dos vetores de suporte que formarão um hiperplano separador das classes a fim de classificar entradas novas, não contidas no banco de treino. Essas novas imagens são entradas da etapa *online*.

Na etapa de identificação de expressão facial ou etapa *online*, uma nova imagem de entrada não pertencente ao banco de dados de treino é inserida no bloco de detecção de faces. O algoritmo Viola-Jones deve localizar a face e realizar a estimativa de um ponto base que será referência para análise estatística da nova face em relação ao modelo AAM previamente treinado na etapa *offline*. Aplicando-se o modelo, obtém-se um vetor de características ou vetor de aparência que descreve parametricamente a forma e a textura da imagem de entrada.

Esse vetor de características é remapeado no espaço de características SVM e é atribuída uma das classes de expressão facial para a imagem de entrada.

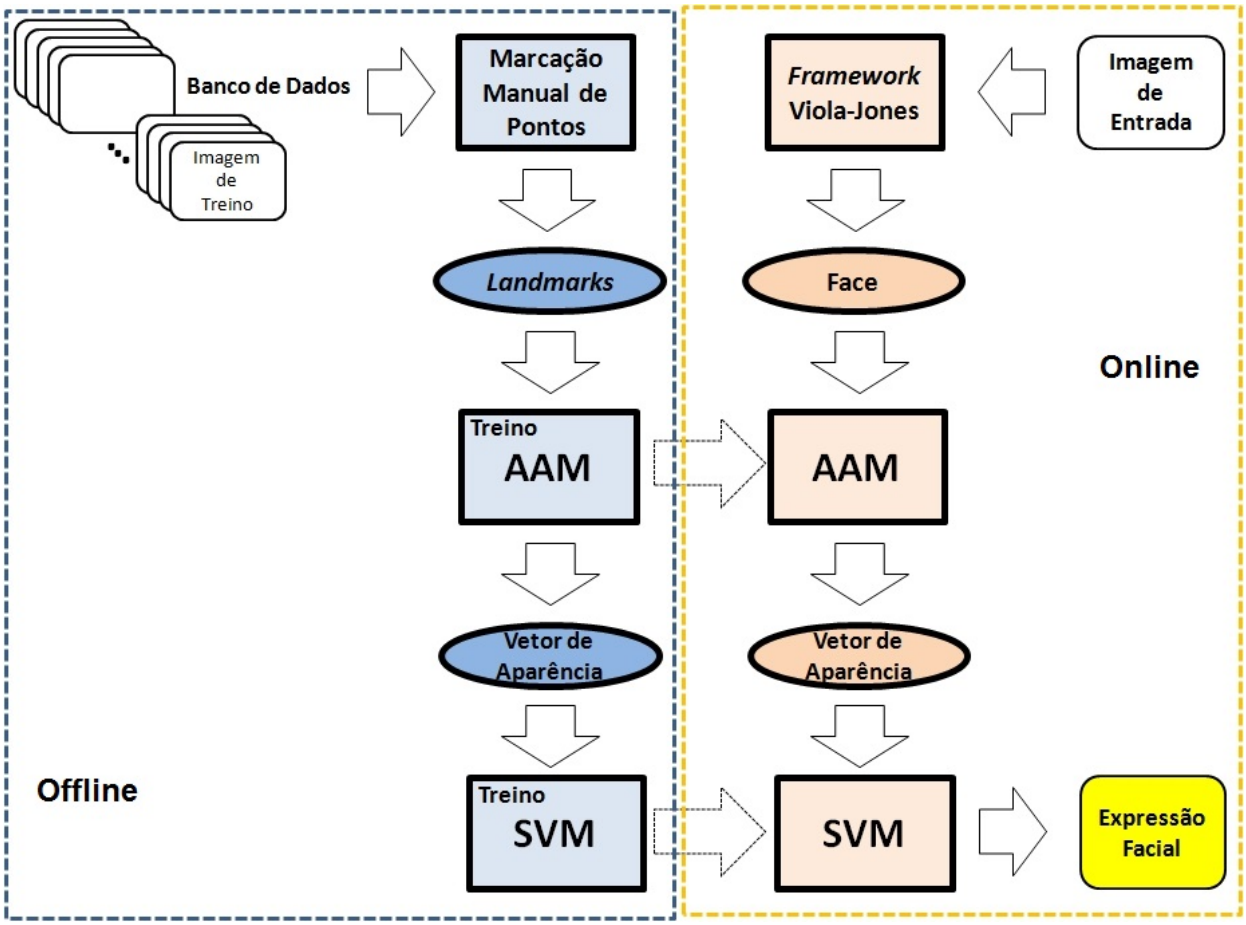


Figura 5.1: Sistema Proposto para a Identificação de Expressões Faciais.

5.1 Banco de Dados

O banco de dados utilizados para o problema de reconhecimento de expressões faciais e modelagem estatística foi o *JAFFE - Japanese Female Facial Expression* (LYONS et al., 1998), apresentado na Figura 5.2. O banco contém um total de 219 imagens de 10 indivíduos femininos de etnia japonesa em pose frontal. Cada indivíduo é apresentado em 3 ou 4 poses para as expressões faciais: neutra, alegria, tristeza, raiva, medo, surpresa e aversão. As imagens estão em escala de cinza em resolução de 256x256 *pixels* com 8 bits por *pixel*.

O motivador da utilização do banco foi seu uso já conhecido na área, possibilitando comparações com trabalhos anteriores. Além disso, suas características de iluminação, *background*,

Figura 5.2: Banco de dados utilizado: *JAFFE - Japanese Female Facial Expression.*

não oclusão e vista frontal permitem bons resultados na localização das faces, acarretando uma análise com menor incidência de erros devido à pré-processamentos do modelo estatístico AAM.

O conjunto de imagens do banco de dados JAFFE foi marcado manualmente com 68 pontos representativos para as expressões faciais. Pontos que compõem o contorno das regiões que envolvem os olhos, boca, nariz, sobrancelhas e queixo foram utilizados. A localização dos pontos segue o modelo estabelecido no Banco de Dados CK+ (KANADE; COHN; TIAN, 2000), conforme Figura 5.3. Esse conjunto de pontos para cada indivíduo é denominado *landmarks*. O banco sofreu alterações em (LUCEY et al., 2010), onde novas sequências de imagens foram adicionadas.

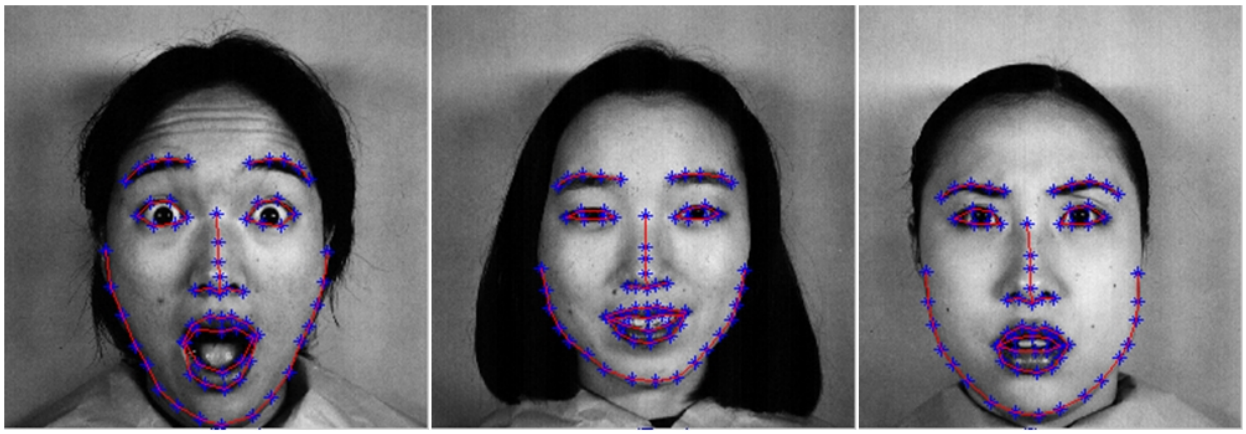


Figura 5.3: Os 68 *landmarks* foram marcados manualmente de acordo com o proposto no banco de dados CK+.

5.2 Detecção de Face

Cada imagem do banco de dados será utilizada tanto para treinamento quanto para teste do sistema através do método de validação *leave one out* (FUKUNAGA, 1990).Portanto é necessário a marcação manual dos *landmarks* para todas as imagens na etapa de treinamento e a correta detecção da localização da face na etapa de teste. O *framework* Viola-Jones foi utilizado nessa etapa e todas as faces das imagens do banco de dados foram corretamente detectadas. Exemplos de sua saída são ilustradas na Figura 5.4.

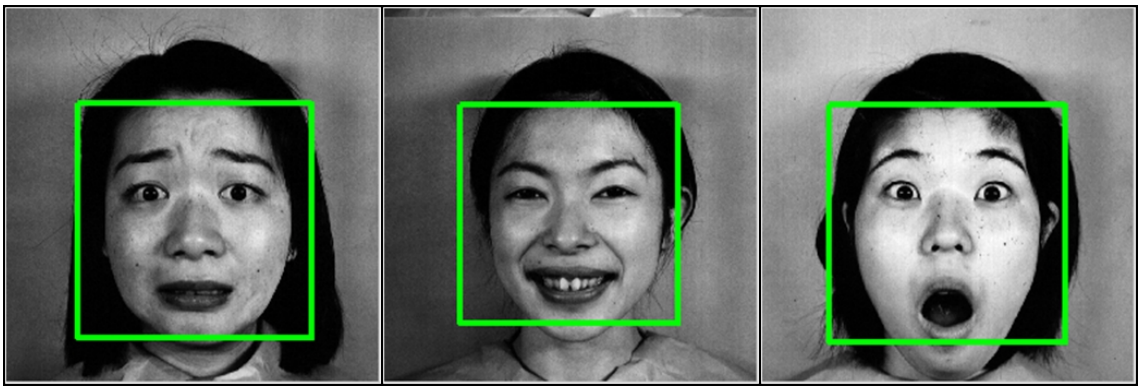


Figura 5.4: O detector de faces Viola e Jones foi capaz de localizar todas as faces para a base de dados JAFFE.

Vale ressaltar que o alto índice de sucesso deve-se, além de um bom algoritmo detector de face, a um banco de dados com imagens em ambiente controlado. O problema de plano de fundo é mínimo e não existem rotações de face e oclusão. Em condições adversas outras técnicas podem apresentar um melhor desempenho. Algoritmos que detectam a face baseados em tom de pele (PASSARINHO C. J. P. ;SALLES, 2012), por exemplo, minimizam o problema de detecção de face em rotação apresentado pelo Viola-Jones.

Com as faces corretamente localizadas, os estágios subsequentes serão favorecidos, pois uma face não localizada acarreta imediatamente em um erro na aplicação do modelo AAM e, consequentemente no classificador.

5.3 Modelamento Estatístico das Expressões Faciais

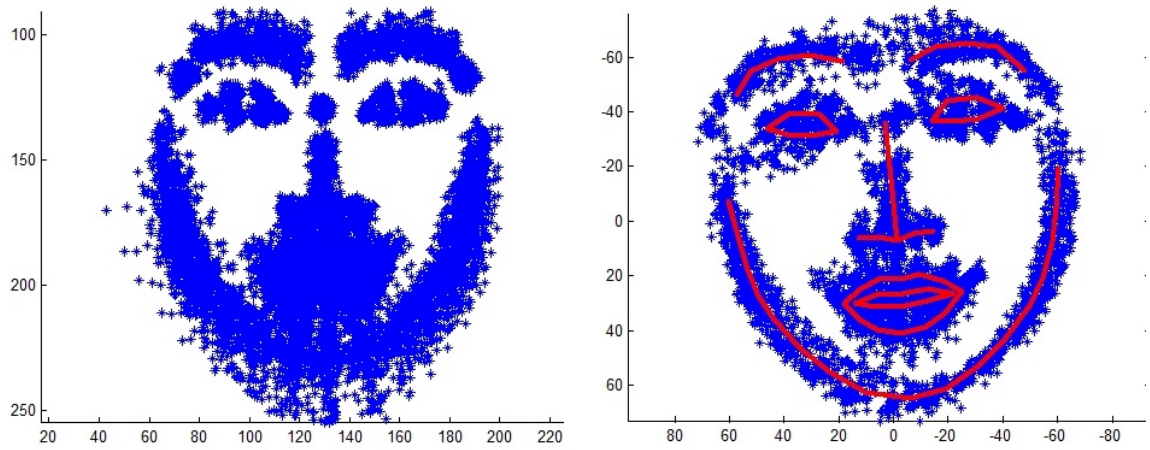
Com os 68 pontos manualmente marcados que constituem o *landmark* para o banco de dados JAFFE é possível analisar a distribuição geométrica e avaliar a forma e textura de cada expressão facial.

As conexões entre esses n pontos bidimensionais, onde $n = 68$ *landmarks*, definem a forma da expressão facial em um vetor de dimensão $2n$. Cada ponto apresenta as coordenadas nos dois eixos do plano cartesiano. Já as áreas delimitadas por um conjunto de três pontos, formando uma triângulo, contém a informação de textura proveniente das intensidades dos *pixels* avaliados dentro dessas áreas.

Portanto, a partir das marcações no treino, é possível gerar um modelo estatístico representativo para as expressões faciais.

5.3.1 Modelo Estatístico de Forma

O modelo estatístico de forma tem como entrada os *landmarks* do banco de dados JAFFE. Para minimizar os problemas de escala, translação e rotação é efetuado a Análise de Procrustes nos dados de entrada, ilustrados na Figura 5.5(a). Observa-se que tal algoritmo é capaz de alinhar as formas de cada expressão facial de treino, propiciando uma melhor avaliação da distribuição estatística dos seus pontos ou *landmarks*. Observa-se ainda a forma da face média em destaque com uma linha contínua, conforme Figura 5.5(b).



(a) (b)

Figura 5.5: (a) Conjunto de dados de entrada para Análise de Procrustes. (b) Alinhamento obtido, minimizando rotação, translação e escala além da forma média para a face na linha contínua.

Obtendo-se a face média, é realizado o treinamento AAM de forma onde realiza-se uma parametrização dos dados de treino. Existe uma deformação b_s capaz de levar cada uma das expressões de entrada para a face média obedecendo a relação $x \approx \bar{x} + P_s b_s$, conforme Equação 3.3.

Para tal, é necessário a análise de componentes principais (PCA) e, portanto, deve-se escolher os z maiores autovalores que representam a forma fielmente, mas respeitando a relação de compromisso entre o número de componentes e erro $x - (\bar{x} + P_s b_s)$. Quanto maior o número de componentes mais x se aproxima de $\bar{x} + P_s b_s$ com consequente elevação do custo computacional para todas as etapas seguintes, em um efeito cascata. Já um número de componentes reduzidas pode gerar um erro de aproximação impraticável, distorcendo as informações de entrada.

Para o cálculo dos z autovalores representativos para aproximação apropriada, levantou-se a curva de contribuição normalizada acumulada C_a dos primeiros j autovetores conforme:

$$C_{a(j)} = \frac{\sum_{i=1}^{j} \lambda_i}{\sum_{i=1}^{n} \lambda_i}. \tag{5.1}$$

A contribuição normalizada de cada componente é apresentada na Figura 5.6. Com apenas os 20 primeiros autovalores e seus autovetores associados desse conjunto de treino é possível armazenar 98% da informação contida se considerado a totalidade dos autovetores.

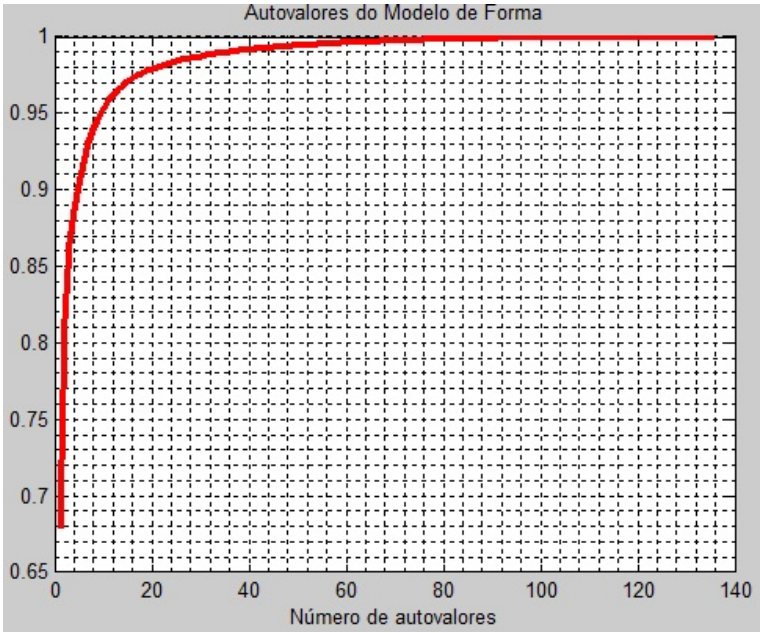


Figura 5.6: Os 20 primeiros autovetores contém 98% da informação total de todos autovetores. Dessa forma, é possível redução de dimensionalidade sem perda de especificidade.

Portanto, faz-se $z = 20$ e com essas 20 componentes ou modos é possível representar a forma de uma expressão facial. Na Figura 5.7 é visualizada a forma média e a influência que os 5 primeiros autovetores exercem sobre ela, deformando-a e levando a uma nova forma conforme Equação 3.4. Verifica-se que o primeiro autovetor está associado à rotação da face,

o segundo muda predominantemente a forma da sobrancelha ao passo que o terceiro altera a boca. De maneira similar, o quinto autovetor possui uma correlação maior com a expressão de raiva enquanto terceiro possui maior correlação com a expressão de surpresa. Ou seja, os vetores de forma tendem a armazenar informações relativa às expressões de modo a ser uma boa entrada para um bloco de classificação.

Figura 5.7: Em vermelho a forma média e em azul a influência dos primeiros 5 autovetores (λ_k) ponderados como $x = \bar{x} \pm P_k 3\sqrt{\lambda_k}$.

A influência dos autovetores é realizada utilizando a expressão $x = \bar{x} \pm P_k 3\sqrt{\lambda_k}$, onde o peso atribuído a cada autovetor não passa dos limites definidos por $\pm 3\sqrt{\lambda_k}$. Esse limite garante que o modelo de forma artificial gerado pelo AAM fica dentro das deformações reais apresentadas pela face humana real (COOTES; EDWARDS; TAYLOR, 1998).

5.3.2 Modelo Estatístico de Textura

Uma vez definido um modelo para a forma é possível utilizar os parâmetros de forma b_s para treino de um classificador. No entanto, uma descrição mais detalhada de uma expressão facial pode ser obtida através da avaliação de sua textura. Isso é possível utilizando-se os *landmarks* como base para a triangulação de Delaunay, conforme Figura 5.8

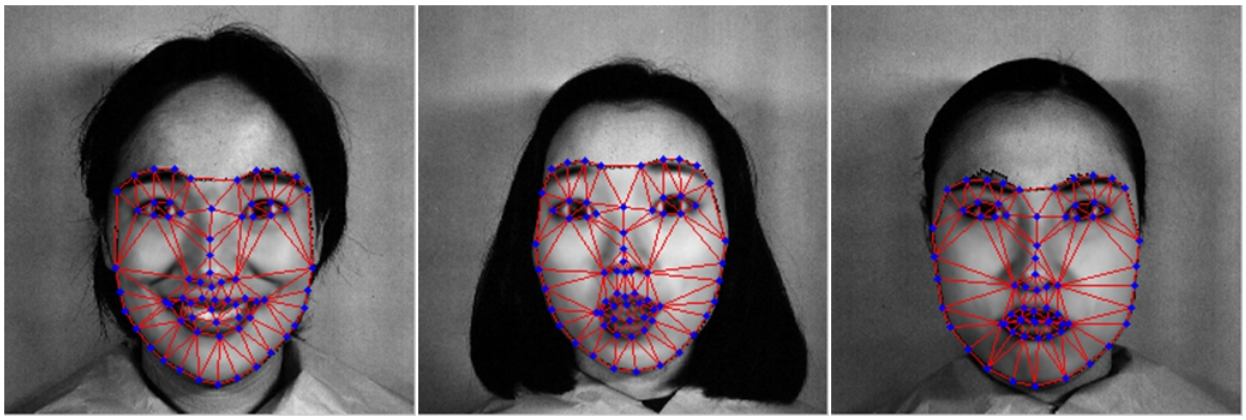


Figura 5.8: Conjunto áreas que representam a textura e forma de uma localidade da face utilizando a triangulação de Delaunay.

A informação de textura não deve ser influenciada pela forma que a expressão facial apresenta, portanto utiliza-se um algoritmo de remapeamento onde cada uma das imagens de entrada apresentadas são levadas para a forma média em um processo de deformação ou *warp*. A Figura 5.9 apresenta algumas imagens do banco de dados com as suas respectivas texturas remapeadas segundo a forma média.

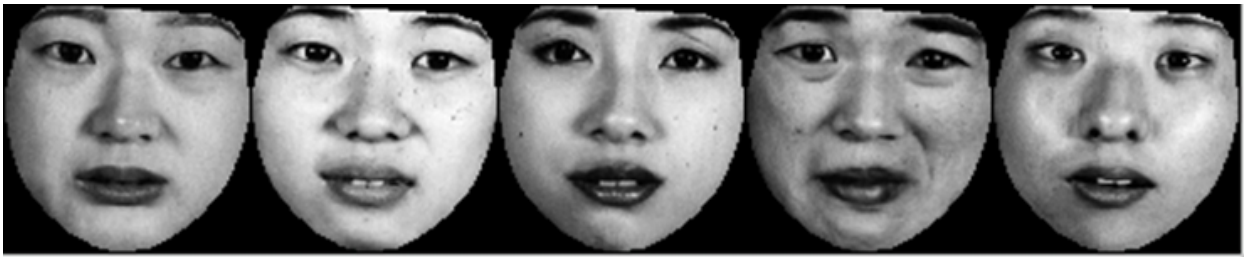


Figura 5.9: Através dos pontos mapeados pela triangulação de Delaunay é possível gerar um conjunto de treino independente da forma, adequado para o modelo estatístico de textura.

O último passo antes de verificar a representação estatística de textura é garantir que as imagens estão alinhas fotometricamente, minimizando os efeitos que diferentes condições

de iluminação podem oferecer para as imagens de entrada. A Figura 5.10 apresenta os resultados da normalização fotométrica utilizando a Equação 3.5.

Figura 5.10: Texturas dos protótipos de treino alinhados fotometricamente após normalização.

Observe que as imagens de saída apresentam intensidade de *pixels* semelhantes, indicando que as imagens estão adequadas para criação do modelo estatístico de textura.

A textura das imagens levadas à forma média pode ser avaliada segundo a relação $g \approx \bar{g} + P_g b_g$. Ou seja, a textura pode ser descrita por um vetor de textura b_g, conforme Equação 3.7. Novamente é necessária a análise de componentes principais e avaliação de quantos autovetores são capazes de representar fielmente a textura de uma expressão facial. Para isso, levanta-se novamente a curva de contribuição C_a em conformidade com a Equação 5.1. A Figura 5.11 ilustra esse processo.

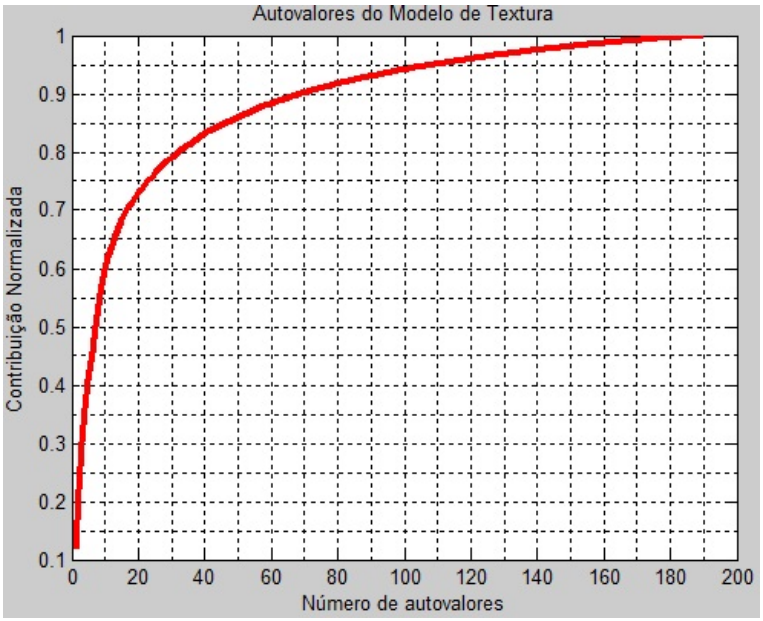


Figura 5.11: Aproximadamente os 30 primeiros autovetores contém 80% da informação total de todos autovetores. Dessa forma, é possível redução de dimensionalidade sem perda de especificidade.

Apenas os 30 primeiros autovetores, responsáveis por aproximadamente 80% da informação de textura, são utilizados após a análise de componentes principais. Esse valor foi obtido experimentalmente, onde verificou-se que um acréscimo de componentes não era capaz de gerar modelos sintéticos mais representativos nem melhorar os índices dos classificadores, perdendo ainda a contribuição de redução de dimensionalidade. Portanto, trabalhar com um número de componentes não é justificável devido ao acréscimo do custo computacional.

Qualquer textura de entrada pode ser vista como uma combinação linear do modelo de textura médio e um somatório ponderado dos autovetores de P_g. O modelo $\bar{g}$ representa a forma média dos pontos com a textura média. A textura média e os primeiros 5 autovetores estão ilustrados na Figura 5.12.

Observa-se que existe a presença de informação sobre as expressões faciais em cada componente. Por exemplo, o primeiro autovetor ajusta a abertura da boca tendendo a expressão de felicidade ao passo que o terceiro autovetor contribui com a expressão de surpresa. No entanto, verifica-se ainda a concentração da informação de identidade de indivíduos específicos na contribuição desses principais autovetores. Ou seja, cada autovetor está associado a uma textura que remete a uma identidade específica, o que não é interessante, pois a informação relevante é a expressão facial independente da identidade do protótipo. Esse efeito pode ser minimizado utilizando-se um banco de dados com maior diversidade de indivíduos.

5.3.3 Modelo Estatístico Combinado de Forma e Textura

Com informações de forma e textura pode-se obter um descritor mais completo de uma expressão facial e utilizar as variáveis de parametrização b_s e b_g como entradas para um classificador. No entanto, existe uma correlação entre forma e textura, uma vez que alterada uma forma as intensidades dos *pixels* também são alteradas, modificando textura. Para serem avaliados os dois parâmetros é calculado a relação W com unidade de $\frac{intensidade\ de\ pixel}{deslocamento\ de\ pixel}$, conforme Equação 3.8.

Um modelo estatístico de aparência que combina forma e textura é obtido avaliando os espaços formados pela junção dos subespaços dos autovetores de forma P_s e autovetores de textura P_g. Esse novo espaço P_c é resultado da análise de componentes principais da união dos subespaços. Os pesos atribuídos a cada autovetor do espaço P_c é dado pelo vetor de aparência c.

Com 22 autovetores é possível combinar forma e textura mantendo aproximadamente 95%

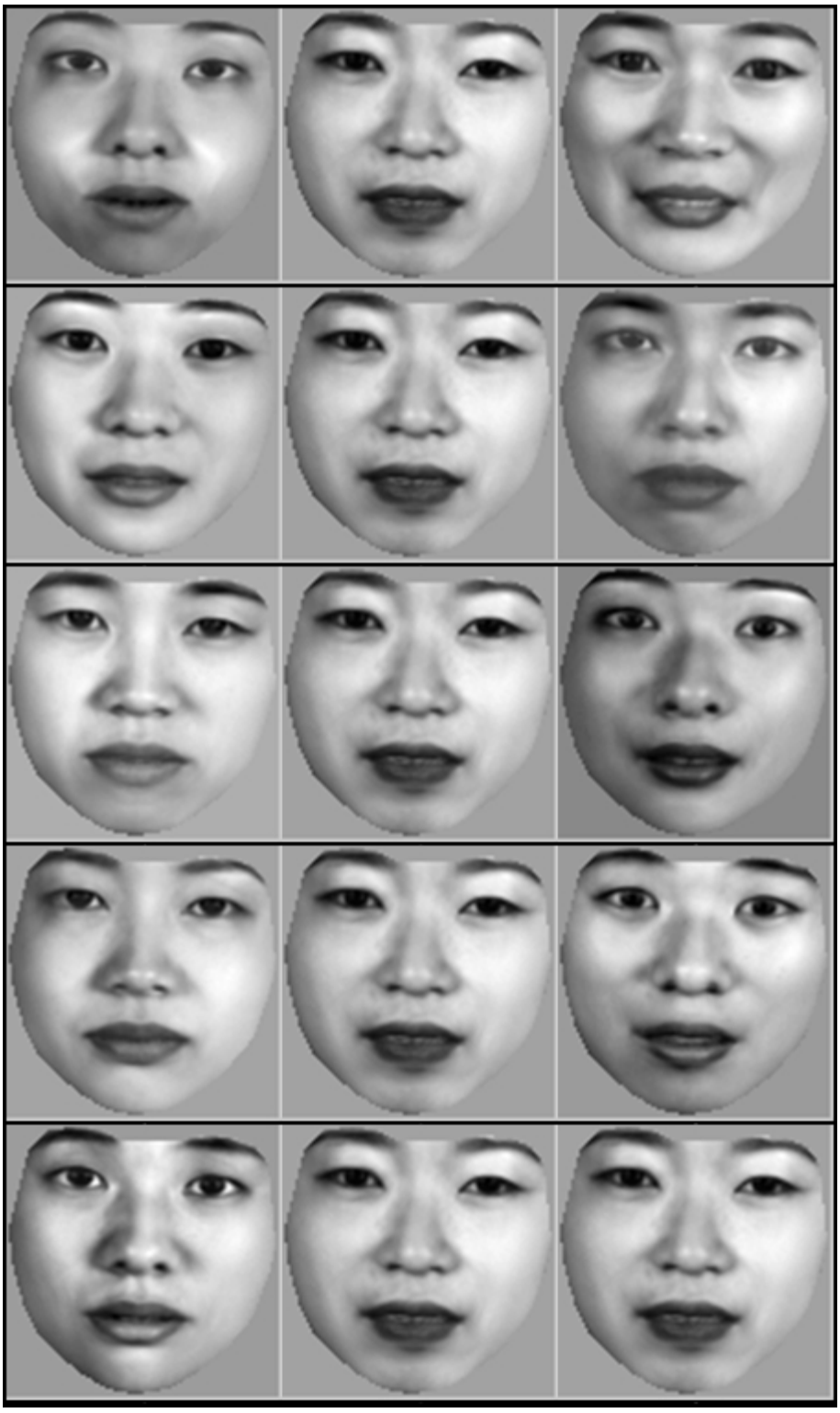


Figura 5.12: Textura média na coluna central e nas colunas 1 e 2 de cada linha k a influência dos primeiros 5 autovetores (λ_k) ponderados como $g = \bar{g} \pm P_k 3\sqrt{\lambda_k}$

da informação proveniente dos dois treinamentos. A Figura 5.13 ilustra o processo de decisão do número de autovetores escolhidos. Observa-se que a contribuição de alguns autovetores é

muito pequena, confirmando que existe correlação entre forma e textura, sendo desnecessário trabalhar com a ambiguidade.

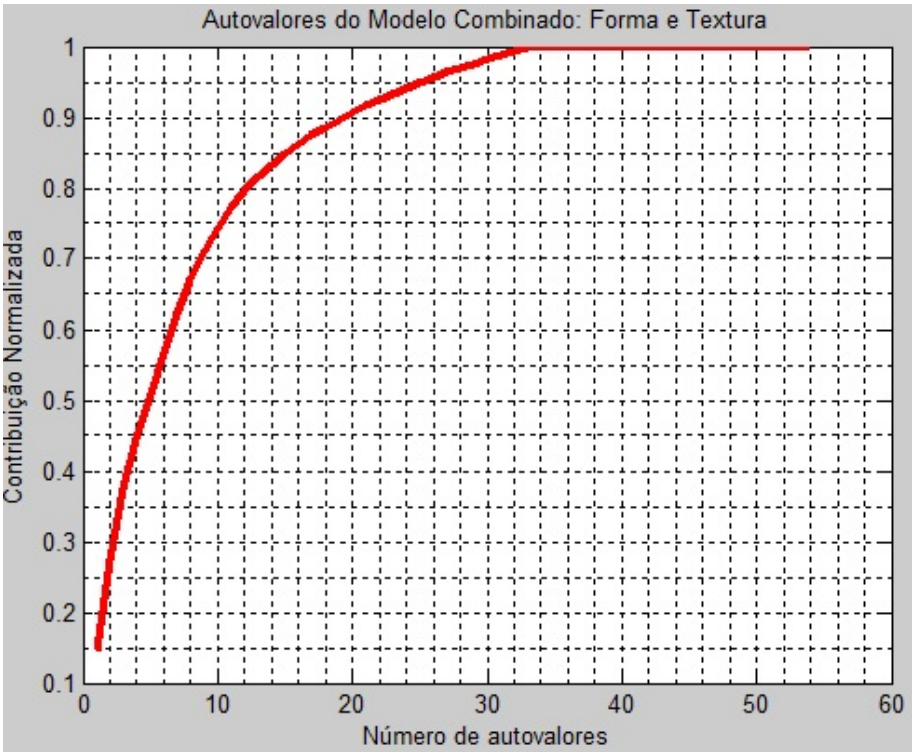


Figura 5.13: Aproximadamente os 22 primeiros autovetores contém 95% da informação total de todos autovetores

A diferença quadrática gerada entre o modelo de aparência e o modelo combinado é mostrada na Figura 5.14. Observe que a perda de informação ocorre em alta frequência, isto é, as regiões de borda são suavizadas e imperfeições na pele como pintas são raras. A imagem apresenta uma aparência suavizada. Portanto, há uma nova redução de dimensionalidade sem acarretar perda considerável de informação que poderá gerar erros no bloco de classificação. A expressão facial e traços da identidade do indivíduo são claramente preservados.

5.3.4 Modelo Estatístico de Busca: Aplicação do Modelo AAM

Com o treinamento do estatístico AAM anterior é possível gerar um modelo sintético para a expressão facial contendo a textura média de uma expressão inserida no molde da forma média. Outras imagens sintéticas podem ser geradas através de manipulações no vetor de aparência c, em concordância com as Equações 3.15 e 3.14. Em um caso dual, onde apresentada uma imagem de entrada é necessário o cálculo do vetor de aparência que compõe aquela face a partir do modelo de treino, é necessário um algoritmo de busca iterativo capaz

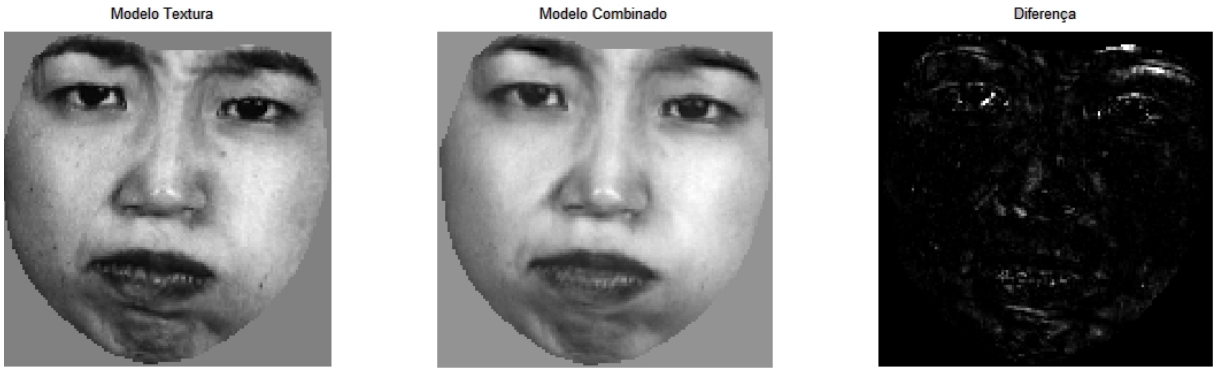


Figura 5.14: Modelo de aparência, modelo combinado e diferença quadrática.

de convergir o modelo de treino para uma imagem sintética representativa para a imagem de entrada.

Além de alterar o vetor de aparência para levar a forma e textura média do modelo obtido no treinamento AAM é necessário possíveis ajustes de escala, translação e rotação. Esses parâmetros combinados formam o vetor de pose $r_{(p)}$. Atualiza-se o vetor de pose através da relação $\delta_p = -Rr_{(p)}$ (Equação 3.27).

Seria necessário recalcular R a cada passo, mas como a busca se inicia sempre com a mesma condição inicial, i.e. modelo sintético médio de forma e textura, aproxima-se R por uma constante. Para os s protótipos de treinamento já conhecidos (imagens do JAFFE e *landmarks* associados) é realizada variações de 0,5 vezes o desvio padrão dos vetores de aparência, 10% de variação em escala, translação de 3 pixels e 10% de variação na textura. Esses protótipos e suas versões escalonadas, transladadas e rotacionadas são a entrada para cálculo da matriz de regressão R utilizando-se a relação $R = \left(\frac{\partial r}{\partial p}^T \frac{\partial r}{\partial p}\right)^{-1} \frac{\partial r}{\partial p}^T$ e o filtro Gaussiano proposto na Equação 3.31.

5.3.5 AAM Multi Resolução

Para uma maior precisão e melhor convergência, o algoritmo AAM utiliza multi-resolução de quatro camadas. Todo o processo de treinamento de modelo de forma, textura, combinado e busca é realizado nas escalas de 0,25 , 0,5 , 0,75 e 1 conforme Figura 5.15. Portanto existem 4 treinamento AAM distintos.

Durante o processo de aquisição de dados em uma imagem de entrada é realizada a busca da

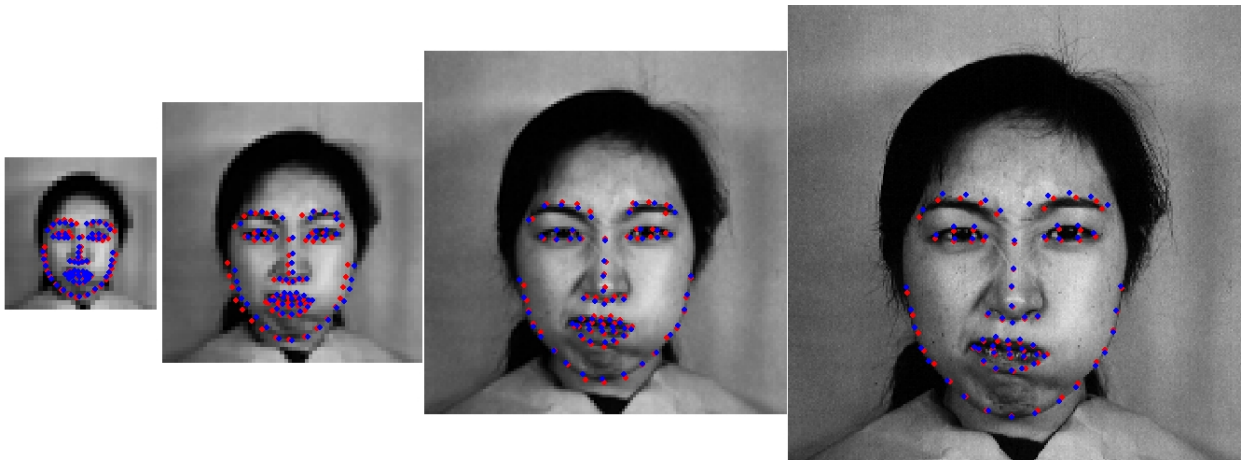


Figura 5.15: A busca pelo modelo sintético é realizado em 4 escalas: 0,25, 0,5, 0,75 e 1.

menor para a maior escala. Dessa forma obtém-se ganho de convergência. A primeira escala do modelo é responsável por direcionar os parâmetros de escala, translação e rotação. Nas escalas subsequentes já existe uma estimativa da forma da imagem de entrada e espera-se a convergência da textura.

O resultado da precisão do modelo sintético gerado para uma imagem de entrada não prevista na etapa de treinamento pode ser aferido através da diferença entre a imagem de entrada e o modelo sintético gerado $\delta I = I_i - I_m$ (Equação 3.16).

5.3.6 Proposta para Convergência do Modelo

Como a precisão da avaliação do erro é altamente dependente do ponto inicial no algoritmo de busca e dado que o processo de busca converge rapidamente, é proposto um conjunto de 9 pontos distribuídos na vizinhança do ponto definido como face pelo bloco de detecção de faces conforme Figura 5.16. Portanto, o centro de massa da área definida como face pelo *framework* Viola-Jones com uma translação de 35 *pixels* no eixo vertical é utilizada como ponto central de busca. O ponto inicial, portanto, aproxima-se da ponta do nariz.

Experimentalmente, observou-se que um deslocamento de 3 ou até mesmo 2 *pixels* em relação ao ponto inicial definido na saída do bloco de localização de face para a aplicação do modelo de busca resulta em uma imagem que não converge corretamente, evidenciando a necessidade de um ponto inicial de busca adequado.

É gerada uma imagem sintética para cada um dos j pontos de teste e associado a cada uma

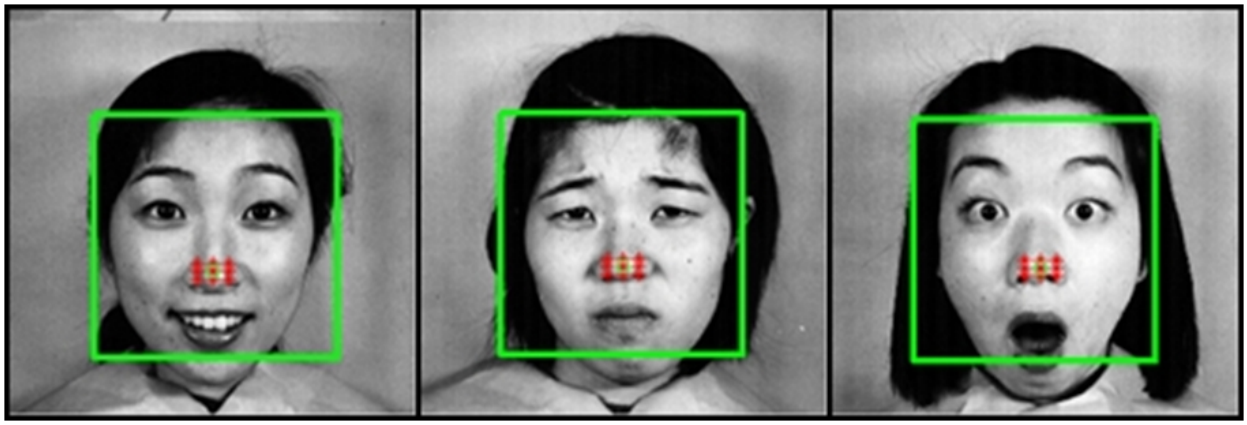


Figura 5.16: É proposto um conjunto de pontos na vizinhança do ponto definido como face pelo bloco de detecção de face para aplicar o modelo de aparência no algoritmo de busca visando minimizar o erro entre a imagem de entrada e o modelo gerado.

tem-se o erro δI_j. O ponto com menor erro apresentado é escolhido como referência e a imagem sintética a ele associada é determinada como a melhor estimativa para sintetizar artificialmente a imagem de entrada. Esse modelo estatístico contém um vetor de aparência c inerente à apresentação de sua expressão facial.

A Tabela 5.1 apresenta o resultado para a escolha do ponto inicial de busca que minimiza o erro do algoritmo AAM. Observa-se que pode-se aproximar os resultados para uma distribuição uniforme com desvio padrão de 4,612%. Ou seja, apenas em 11,03% dos casos o ponto inicial de busca definido pelo *framework* Viola-Jones foi mantido. Nos 88,97% restante dos casos o método proposto diminui o erro no modelo sintético final.

# Ponto	**Taxa de Escolha** [%]
1	11,99
2	11,39
3	11,50
4	10,55
5 (ponto central)	11,03
6	10,70
7	10,78
8	10,81
9	11,24

Tabela 5.1: A distribuição na escolha dos pontos que diminuem o erro gerado pelo AAM baseado na média de 10 repetições pode ser aproximada por uma distribuição uniforme com desvio padrão de 4,612%.

A Figura 5.17 exibe a convergência do modelo de aparência treinado para um modelo que

se aproxima da imagem de entrada.

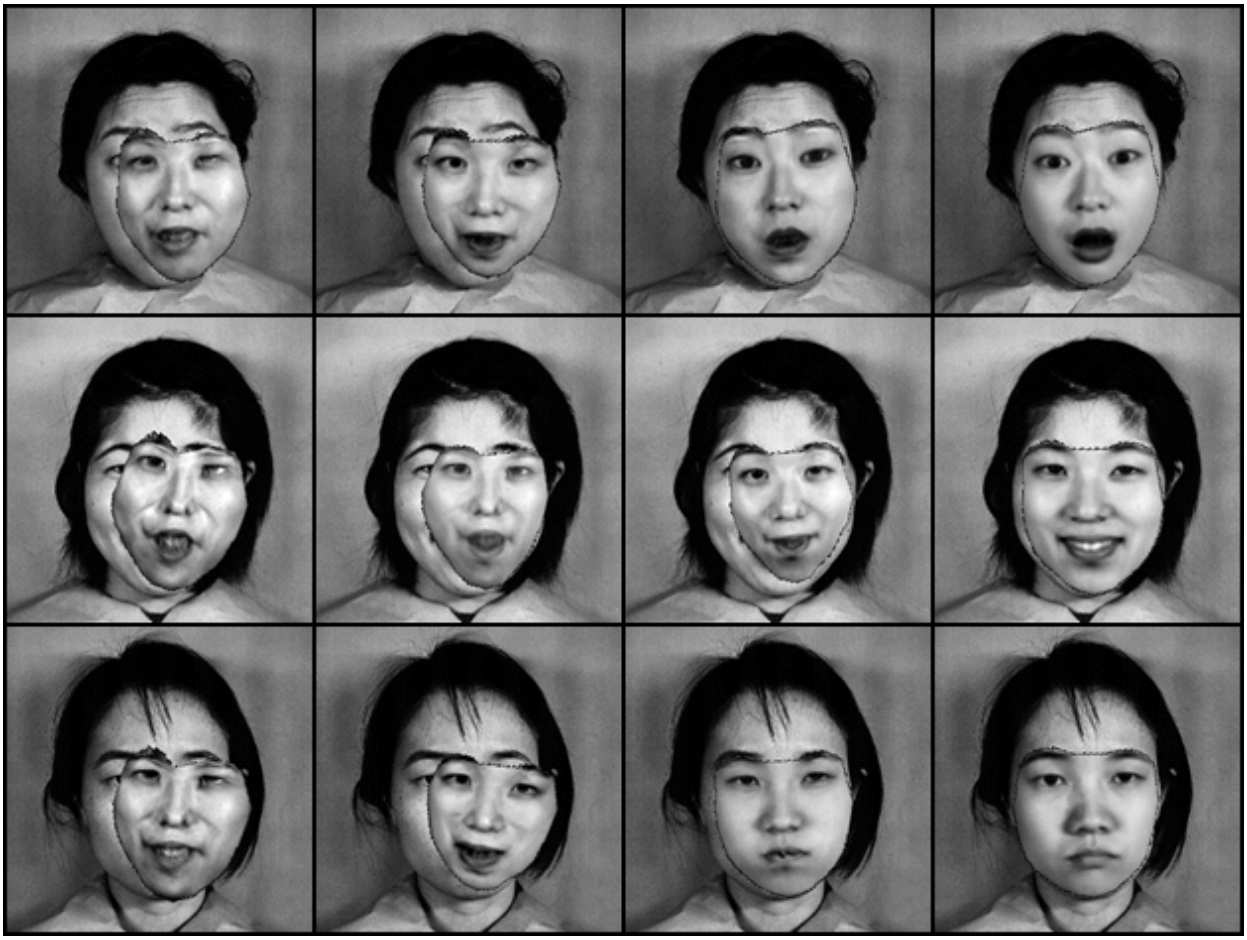


Figura 5.17: Resultado da convergência do modelo treinado de aparência para a imagem de entrada. Na primeira coluna é mostrado o resultado da primeira iteração de busca a partir do modelo médio treinado.

Outra proposta do trabalho é uma modificação da Equação 3.32 de atualização dos pesos.

Originalmente, a cada iteração o vetor de pose e aparência associados $p = [c^T \ t^T]$ é atualizado conforme $\hat{p} = p + k\delta p$, variando-se k. Consequentemente, o vetor de aparência que define a forma e textura do modelo sofre uma atualização linear da forma $\hat{c} = c + k\delta c$. Portanto, todos os pesos de c que influenciam na importância dos autovetores associados ao espaço definido por P_c e que compõem a expressão facial são incrementados com mesma proporção/intensidade. Isso garante iterações controladas, mas é destoante no que se refere ao comportamento esperado não linear mais próximo da realidade, pois é inerente ao processo de análise de componentes principais o fato de que os primeiros autovetores projetam os dados de forma a maximizar a variância dos dados em relação aos últimos. Seria de interesse uma abordagem que privilegia as componentes mais significativas, esperando-se que essa modificação resulte em uma melhor convergência. O modelo híbrido de atualização

proposto é definido pelas relações

$$\hat{p} = \begin{bmatrix} \hat{c} \\ \hat{t} \end{bmatrix} = \begin{bmatrix} c \\ t \end{bmatrix} + \begin{bmatrix} (0,6k + 0,4\lambda_c) & k \end{bmatrix} \begin{bmatrix} \delta c \\ \delta t \end{bmatrix}, \tag{5.2}$$

e

$$\hat{c} = c + (0,6k + 0,4\lambda_c)\delta c, \tag{5.3}$$

onde λ_c corresponde aos autovalores associados aos autovetores que definem P_c. Desse forma, uma parte da atualização ocorre de forma linear e a outra parte de forma não linear, uma vez que os autovalores não são iguais.

Para cada iteração de busca realiza-se a atualização dos parâmetros p através da equação de atualização original e com a equação proposta. A atualização que apresenta o menor erro obtido é mantida. Essa modificação resultou na diminuição do erro δI em 12% das imagens de teste.

5.4 Classificação

Uma vez localizada a face, treinado o modelo estatístico de forma, textura e busca capaz de representar as expressões faciais por meio do vetor de aparência c, é possível realizar a classificação de novas imagens de entrada. Essa tarefa exige o treinamento de um classificador para separar as classes de expressões faciais de alegria, tristeza, raiva, medo, surpresa, aversão além da face neutra.

O banco de dados JAFFE foi dividido em grupos de treino e grupos de teste. Nesse processo foi utilizado o método de validação *leave-one-out*. As imagens de apenas um dos indivíduos em todas as expressões faciais foram separadas como grupo de teste a ser classificado utilizando os parâmetros de classificação obtidos com o treinamento do AAM e do classificador com as imagens dos outros nove indivíduos e seus respectivos rótulos ou classes. Portanto, cada treinamento foi realizado 10 vezes tanto para a geração dos modelo estatísticos de forma, textura, combinado e busca quanto para o classificador, conforme Figura 5.18.

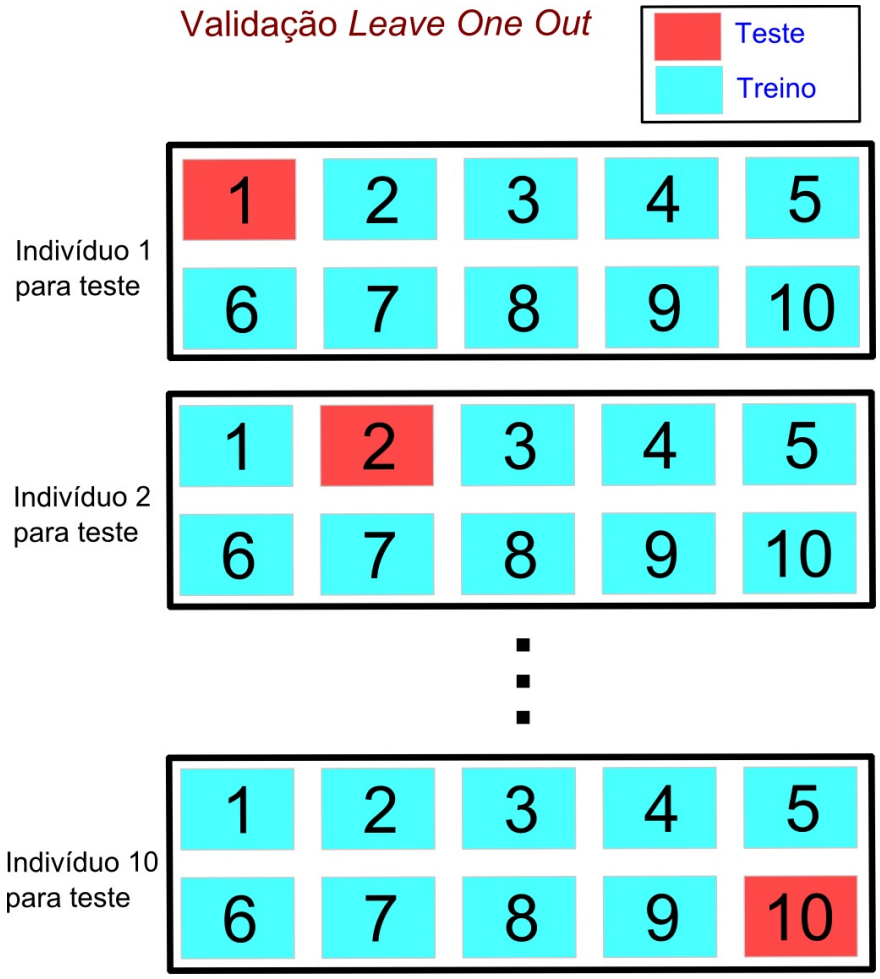


Figura 5.18: Validação *leave-one-out* utilizada para separar os grupos de treino e teste.

A classificação baseada nos vizinhos mais próximos (k-NN) foi o primeiro classificador a ser testado. Sua simplicidade facilita a verificação da capacidade discriminante do modelo estatístico AAM e posterior comparação com um método mais sofisticado de separação de classes escolhido como a máquina de vetores de suporte.

5.4.1 k-NN

Em um primeiro teste de classificação realizado, foram utilizadas apenas as faces neutras, totalizando 30 imagens. Para cada iteração de teste uma das imagens foi rotulada como imagem de teste e as outras 29 foram utilizadas nos treinamento para os modelos de busca e combinado AAM, totalizando 30 treinamentos e classificações distintas. Portanto, o teste consiste em verificar a capacidade do AAM em representar fielmente uma face neutra, operando como um identificador de indivíduos com faces neutras. Dessa forma, o primeiro teste

apresenta 10 diferentes classes.

Foi alcançada uma taxa de acerto de 100% utilizando o classificador com os três vizinhos mais próximos (3-NN) utilizando a métrica de Distância Euclidiana. A escolha de três vizinhos foi obtida com o menor erro de validação cruazada para o 1-NN e 3-NN. Um número maior de vizinhos não é plausível porque cada indivíduo apresenta apenas três faces neutras. Como uma das faces é utilizada para testes a escolha da classe correta é realizada utilizando as outras duas faces rotuladas no grupo de treino. Portanto, a taxa de acerto demonstra que a classificação correta implica no cálculo das distâncias entre as duas faces de mesma classe da imagem de entrada e da própria imagem de entrada a serem inseridas no grupo dos três vizinhos mais próximo. O terceiro vizinho, consequentemente, pertence a uma das outras classes.

Esse resultado indica que o modelo AAM gerado apresenta uma boa representatividade para classes neutras já que um classificador simples separou corretamente a totalidade de classes e é relevante, haja vista que se for possível remapear uma expressão facial qualquer em uma face neutra, pode-se obter alta taxa de acerto em um sistema de identificação de indivíduos que apresente como entrada indivíduos em qualquer expressão facial.

A Figura 5.19 ilustra as faces neutras sintéticas obtidas no treinamento e que foram corretamente classificadas.

Figura 5.19: Faces neutras utilizadas prara o reconhecimento de indivíduos utilizando o 3-NN com validação *leave-one-out*.

O próximo teste consistiu de verificar a separabilidade de uma expressão facial qualquer para a expressão neutra. Nesse momento não há mais o interesse em identificar o indivíduo e portanto, imagens de diferentes indivíduos, mas com mesmas expressões faciais pertencem

a mesma classe.

Foram realizados 7 blocos de testes onde a cada passo foram utilizadas a expressão neutra em conjunto com mais uma expressão do grupo alegria, tristeza, raiva, medo, surpresa e aversão. Para cada bloco uma imagem foi separada como teste sendo a entrada do modelo de busca AAM e do classificador treinados com as imagens restantes. Foi empregado o método de validação *leave one out*.

Novamente, utilizou-se o 3-NN. A escolha de $k = 3$ foi realizada após testes com $k = 1$, $k = 3$, $k = 5$, $k = 7$ e $k = 9$, sendo o valor escolhido o que gerou menores erros de classificação na validação cruzada. Os resultados evidenciados na Tabela 5.2 demonstram que é possível separar a face neutra de uma face não neutra, motivando o acréscimo de novas expressões.

Expressão Facial	**Taxa de Acerto** [%]
Neutra + Raiva	86,67
Neutra + Nojo	79,96
Neutra + Medo	70,97
Neutra + Felicidade	80,33
Neutra + Tristeza	85,25
Neutra + Surpresa	80,00

Tabela 5.2: Resultados do 3-NN para Face Neutra e outra expressão não neutra.

A maior taxa de acerto foi de 86,67% obtida para o conjunto de teste utilizando as expressões neutra e raiva. Em seguida tem-se o conjunto neutra e tristeza com 85,25% seguido do conjunto formado pelas expressões neutra e felicidade com 80,33%. A pior taxa de acerto foi para o conjunto neutra e medo com 70,49%. Em média, a face neutra foi separada de uma das outras expressões faciais para 80,53% dos casos. Os dados indicam que é possível separar as expressões faciais utilizando o modelo AAM e um classificador que tem o vetor de aparência como parâmetro de entrada.

Os resultados obtidos encorajaram o acréscimo de outras expressões. Os novos resultados estão sumarizados na Tabela 5.3. Como esperado, observa-se que o acréscimo de composições com três ou mais expressões faciais resultam em menores taxas de classificação, pois aumenta-se o número de classes e elementos descritivos com sobreposição. A boca fechada, por exemplo, aparece nas expressões de raiva e tristeza. Isso indica a presença de correlação de elementos de classes distintas que são mapeados nos mesmos parâmetros do vetor de características.

Expressões Utilizadas	Taxa de Acerto [%] 3-NN
N+R	86,67
N+Nj	79,96
N+M	70,97
N+F	80,33
N+T	85,25
N+S	80,00
N+F+T	73,91
N+F+T+S	65,57
N+F+T+M	62,10
N+F+T+M+Nj	41,18
N+F+T+S+M	57,79
N+R+F+T+S+M+Nj	47,42

Tabela 5.3: Resultados para classificação das expressões neutra (N), raiva (R), nojo (Nj), medo (M), felicidade (F), tristeza (T) e surpresa (S) utilizando k-NN

Ainda sim é possível encontrar taxas de acerto superiores a 65% para o caso de quatro classes como na composição Neutra + Felicidade + Tristeza + Surpresa. A composição Neutra + Felicidade + Tristeza + Medo alcançou 62,10% de acerto.

Observa-se que há uma clara confusão entre as várias expressões quando inseridas em meio a outras expressões. Mesmo sendo separáveis quando tomadas isoladamente, por exemplo, as classes neutra e tristeza apresentam forte correlação quando estão no meio de outras expressões. Ou seja, quando sozinhas a menor mudança de forma ou textura é prontamente refletida no classificador através do vetor de aparência. Já em meio a outras classes, a separação de classes é realizada pelo classificador com pesos maiores a características que não acentuam os descritores capazes de separar essas classes, aumentado a correlação.

A expressão de nojo também apresenta uma forte correlação com a expressão de face neutra. Na formação dessa expressão a deformação de textura e forma é muito pequena se comparado às outras expressões. Logo, a distância entre a expressão de nojo e de face neutra é menor quando comparado com expressões onde ocorrem maiores deformações como na expressão de surpresa onde há grande elevação de sobrancelha e abertura de boca.

Observe na Figura 5.20 a correlação existente entre as expressões de face neutra, tristeza e nojo. Quanto maior a correlação apresentada entre classes, menores são os índice de acerto na classificação, uma vez que o vetor de características de entrada do classificador não consegue separar claramente essas classes das demais.

(a)

(b)

(c)

Figura 5.20: A forte correlação das expressões de (a) face neutra, (b) tristeza e (c) nojo em meio a um conjunto maior de classes reflete em uma menor taxa de acerto.

Uma maneira de verificar os resultados em relação à questão de separabilidade de classes pode ser obtida através da análise dos dados projetados no Mapa de Sammon (SAMMON, 1969) (Ver Apendice C). Os dados extraídos do vetor de aparência c possuem alta dimensão. Através de um remapeamento de Sammon tenta-se projetar os dados em um espaço bidimensional preservando as distâncias entre os dados no espaço original. Portanto, espera-se que quanto maior a facilidade do classificador em separar os dados, mais afastadas as amostras de classes distintas estarão no novo espaço. Observe na Figura 5.21 como a classificação no caso da face neutra e uma expressão facial adicional é viável, ainda que as classes não sejam linearmente separáveis. No entanto, observe na Figura 5.22 como o caso de todas as classes torna o problema de classificação muito mais complexo, uma vez que a distância entre os

dados diminui e há sobreposição de classes.

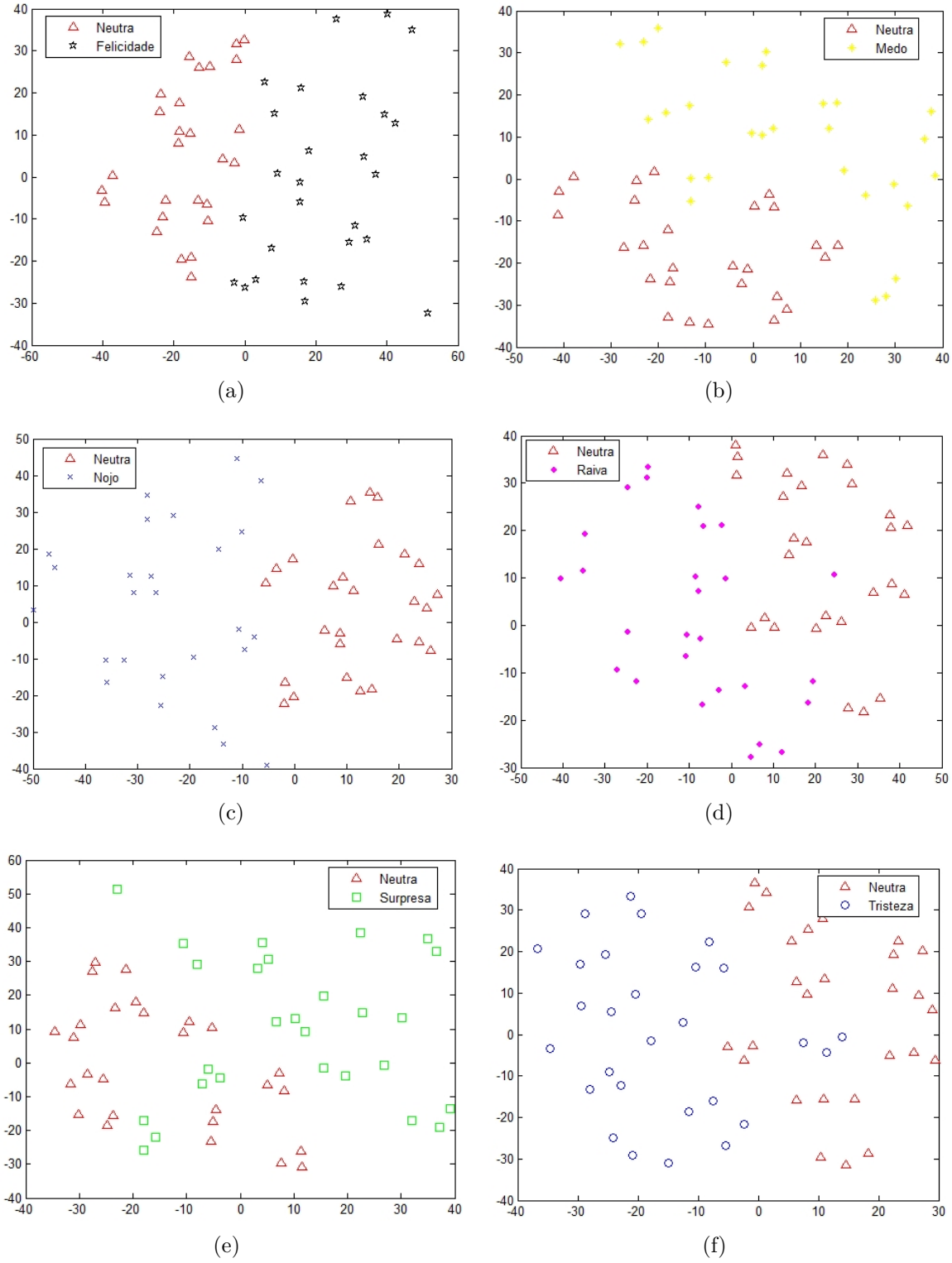


Figura 5.21: Projeção dos dados no espaço bidimensional de Sammon utilizando a expressão Neutra além das expressões (a) Felicidade, (b) Medo, (c) Nojo, (d) Raiva, (e) Surpresa e (f) Tristeza.

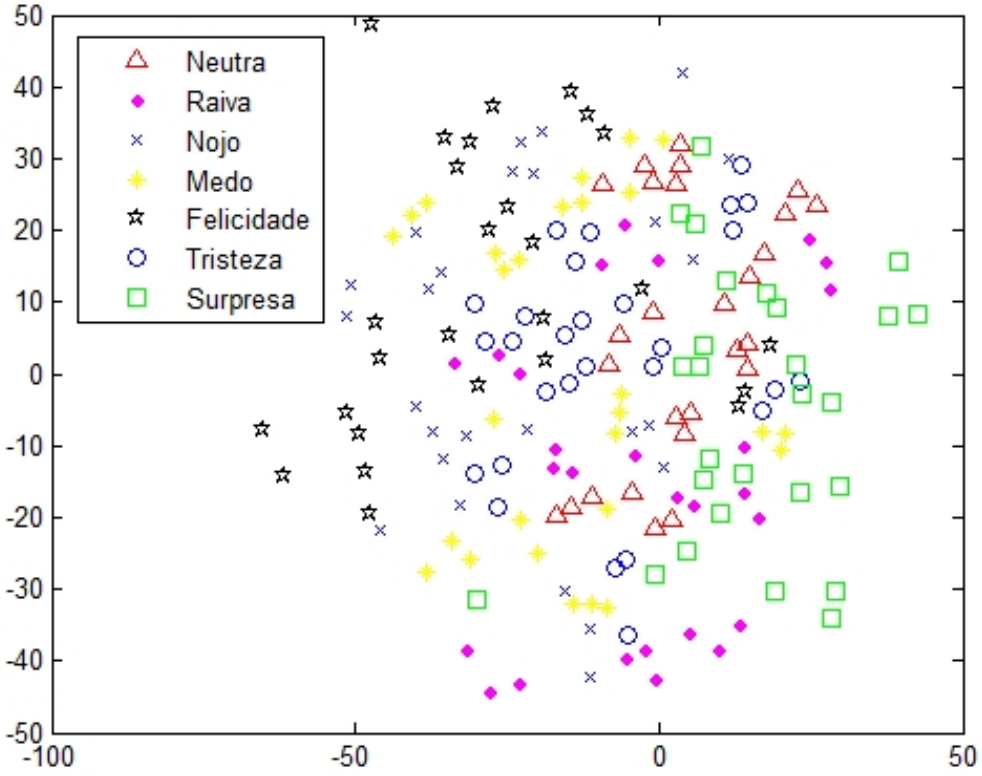


Figura 5.22: A projeção do caso com todas as classes de expressões faciais no Mapa de Sammon evidencia a sobreposição de classes e dificuldade posterior para o bloco de classificação.

5.4.2 Máquina de Vetores de Suporte - SVM

Tendo em vista que o modelo estatístico gerado pelo AAM é representativo, um melhor classificador pode aumentar as taxas de acerto obtidas com o k-NN. O SVM foi escolhido como método alternativo. Sua capacidade de remapeamento em um novo espaço de características impulsiona seu uso como tentativa de separar classes não linearmente separáveis.

A implementação foi realizada utilizando a biblioteca libsvm (CHANG; LIN, 2011).

O kernel escolhido foi o *Radial Basis Function* - RBF. Como opera de forma não linear, $K(\mathbf{x}_i, \mathbf{x}_j) = e^{(-\gamma\|\mathbf{x}_i - \mathbf{x}_j)\|)}$, espera-se que seus hiperplanos separadores sejam mais apropriados para o problema não linear de separar as expressões faciais do que separadores lineares como, por exemplo, utilizando o kernel polinomial. Além disso, as escolhas corretas de seus parâmetros podem acarretar resultados semelhantes ao obtidos com os kernels gaussiano ou polinomial (KEERTHI; LIN, 2003).

Para os testes, novamente foi utilizada a validação *leave one out*. Em cada um dos dez testes realizados, um indivíduo tinha suas expressões separadas como dados de testes a

serem entradas para o extrator de características AAM e classificador SVM treinado com as expressões dos outros nove indivíduos.

O parâmetro do *kernel* γ e o parâmetro C (vide Seção 4.2.2) associado à minimização do erro durante o treinamento do classificador foram obtidos através de uma validação cruzada utilizando o *v-fold* para o grupo de treinamento. Portanto, o grupo de treinamento é divido em v subgrupos e a cada iteração um dos subgrupos é testado utilizando os dados do classificador treinado com os outros $v-1$ subgrupos. Essa validação cruzada é importante para minimização do problema de *overfitting* ou sobre ajuste, onde o classificador consegue separar bem as classes apresentadas no treinamento, mas não é capaz de lidar com a generalização, ou seja, não classifica bem novos dados de entradas desconhecidos.

A busca para os melhores parâmetros γ e C foram feitas realizando buscas exaustivas em uma área maior com um determinado passo de busca. Uma vez localizada o ponto que apresenta melhor desempenho, uma nova busca é iniciada centrada nesse ponto com um passo de busca em menor escala. Por exemplo, inicia-se $\gamma = [1 \quad 2 \quad 3 \quad 4 \quad 5]$ e verifica-se que $\gamma = 4$ apresenta o melhor resultado. O valor de γ final é obtido na busca no intervalo definido por $\gamma = [3,8 \quad 3,9 \quad 4,0 \quad 4,1 \quad 4,2]$.

Outro ponto importante a ser destacado é a normalização dos vetores de aparência que são responsáveis pelo treinamento do SVM. O vetor de características tem dimensão variável entre 24 e 26 dependente do grupo utilizado para o treinamento para o caso de todas as classes. Nesses casos, é comum algumas posições do vetor apresentarem valores numericamente maiores em módulo do que outras. Isso pode ser problemático para o classificador, pois corre-se o risco de atribuir-se implicitamente pesos maiores para algumas poucas características que se destacam numericamente. Imagine um classificador que tenha como parâmetros de entrada o número de habitantes de uma residência e o valor do imóvel em R$. Pode-se formar vetores como $\mathbf{x}_1 = [4 \quad 150.000]$, $\mathbf{x}_2 = [1 \quad 120.000]$, $\mathbf{x}_3 = [3 \quad 300.000]$. Como o valor é muito maior em módulo o classificador pode atribuir maior significância para esse dados, ao passo que espera-se atribuir mesma importância para todos os dados.

Os vetores de treino do classificador são normalizados dividindo-se cada posição pelo maior valor em módulo obtido para aquela posição dentre o grupo de treinamento. De mesma maneira, o dado de teste também é normalizado utilizando como base os mesmos valores. No exemplo anterior, o maior número de habitantes é 4 e o maior valor de imóvel 300.000. Dessa forma, os vetores normalizados são $\mathbf{x'}_1 = [1 \quad 0.5]$, $\mathbf{x'}_2 = [0.25 \quad 0.4]$, $\mathbf{x'}_3 = [0.75 \quad 1]$.

Os resultados para o classificador utilizando o SVM são apresentados na Tabela 5.4. Observa-se que a utilização do 2-*fold* na busca pela otimização dos parâmetros do *kernel* apresentou

melhores resultados do que o 5-*fold* para a totalidade de classes.

Expressões Utilizadas	**Taxa de Acerto [%] SVM - RBF 2-fold**	**Taxa de Acerto [%] SVM - RBF 5-fold**
N+R	73,33	81,67
N+Nj	83,05	67,80
N+M	85,48	74,19
N+F	80,33	90,16
N+T	60,66	88,52
N+S	65,00	85,00
N+F+T	52,17	88,04
N+F+T+S	41,80	72,13
N+F+T+M	53,23	82,26
N+F+T+M+Nj	51,63	65,36
N+F+T+S+M	41,56	61,04
N+R+F+T+S+M+Nj	55,40	45,07

Tabela 5.4: Resultados para classificação das expressões neutra (N), raiva (R), nojo (Nj), medo (M), felicidade (F), tristeza (T) e surpresa (S) utilizando o SVM

Para o caso mais complexo envolvendo todas as expressões do banco foi possível obter uma taxa de acerto de 55,4% com sensibilidade 60,25% e especificidade 93,95%, conforme matriz de confusão apresentada na Tabela 5.5.

A sensibilidade é calculada verificando a razão entre o número de verdadeiros positivos obtidos pelo sistema em relação ao conjunto de verdadeiros positivos e falsos negativos. Portanto, é uma medida que avalia a capacidade de detecção de uma determinada expressão facial quando ela está presente, como se observa em

$$Sensiblidade = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos + Falsos Negativos}}. \tag{5.4}$$

Os verdadeiros positivos foram obtidos da Tabela 5.5 utilizando o percentual de acerto por classe. Ou seja, os elementos que apresentam mesma classe na linha e coluna. Os falsos positivos são os elementos de uma coluna com exceção da entrada referente ao verdadeiro positivo. A sensibilidade do sistema foi calculada utilizando o percentual de todas as classes, uma vez que existem diferentes quantidades de imagens para cada classe.

A especificidade é obtida verificando a razão entre o número de verdadeiros negativos obtidos pelo sistema em relação ao conjunto de verdadeiros negativos e falsos positivos. Portanto, é uma medida que avalia a capacidade do sistema em descartar uma determinada expressão

facial quando ela não está presente, sendo escrita como

$$Especificidade = \frac{\text{Verdadeiros Negativos}}{\text{Verdadeiros Negativos + Falsos Positivos}}. \tag{5.5}$$

Os falsos positivos foram obtidos da Tabela 5.5 utilizando o percentual de detecção de uma classe quando a imagem aferida pertencia a uma outra. Ou seja, os elementos de uma linha da coluna com exceção do verdadeiro positivo. Os verdadeiros negativos são os elementos que não são entradas para os verdadeiros positivos ou falsos positivos. Portanto, são todos os elementos que para uma determinada classe foram detectados como outra. A especificidade dos sistema foi calculada utilizando o percentual de todas as classes, uma vez que existem diferentes número de imagens para cada classe. A alta sensibilidade está relacionada com o sistema não possuir nenhuma imagem detectada com ausência de face, reduzindo o número de falsos positivos. Além disso, o número de verdadeiros negativos é muito maior do que falsos positivos por se tratar de um problema multi-classe. Nesse caso, quando avalia-se uma das sete classes, as imagens erroneamente classificada como pertencentes a uma das outras 6 classes reflete no número de verdadeiros negativos.

Esse resultado mostra que o SVM apresenta uma taxa de acerto superior que o k-NN, aumentando em 7,98% o percentual de acerto em relação ao classificador anterior. A expressão de tristeza apresentou menor taxa de detectção ao passo que todas as expressões neutras foram corretamente classificadas.

	Raiva	**Nojo**	**Medo**	**Felicidade**	**Neutra**	**Tristeza**	**Surpresa**
Raiva	0,407	0	0	0,035	0	0,107	0
Nojo	0,037	0,667	0,148	0,103	0	0,214	0
Medo	0	0,222	0,704	0,241	0	0,321	0,074
Felicidade	0,259	0,111	0,148	0,517	0	0,107	0
Neutra	0,259	0	0	0,069	1	0,143	0,111
Tristeza	0	0	0	0	0	0,108	0
Surpresa	0,037	0	0	0,035	0	0	0,815

Tabela 5.5: Matriz de confusão para classificação utilizando todas as expressões faciais do Banco de Dados JAFFE e classificador SVM com *kernel* RBF.

Analisando individualmente cada conjunto de treinamento/classificação, verifica-se que o indivíduo 7 do banco de dados apresentou os piores resultados, uma vez que o algoritmo de busca do AAM não consegue sintetizar corretamente a imagem de entrada para todas as expressões, conforme Figura 5.23. Excluindo-se tal elemento é possível aumentar a taxa de acerto para 0,6010.

Figura 5.23: O indivíduo 7 do banco de dados não foi corretamente modelado pelo AAM, o que gerou vetores de características inapropriados e consequente erro de classificação.

5.5 Desempenho

O sistema proposto realiza a classificação de expressões faciais de maneira automática e será comparado com outras abordagens que utilizaram o JAFFE, apesar de disporem de localização manual ou recorte da face.

Em 1998, Zhang et al. (ZHANG et al., 1998) alcançaram a taxa de 90,10% de acerto utilizando *Wavelets* de Gabor e classificador *perceptron*. A validação realizada foi a validação cruzada *v-fold*. Nesse trabalho foi utilizado um escalamento para fixar a distância entre os olhos e um recorte da imagem (*cropping*), trabalhando-se apenas com uma região de interesse de menor resolução (256×256 *pixels*).

No ano seguinte Lyons et al. (LYONS; BUDYNEK; AKAMATSU, 1999) apresentou resultado de 92% de taxa de acerto utilizando a validação 10-*fold* em um sistema composto por *Wavelet* de Gabor e LDA. Os autores utilizaram um *grid* de 34 pontos ajustados manualmente a fim de extrair características somente das faces.

Dubuisson (DUBUISSON; DAVOINE; MASSON, 2002) apresentou em 2002 um sistema utilizando projeção de características em um subespaço através de PCA em conjunto com LDA e um classificador baseado em uma árvore de decisão por nó. Com validação cruzada conseguiu 87,60% de taxa de acerto utilizando além do JAFFE, outros banco de dados como Yale e CMU-Pittsburgh. As imagens de entrada foram manualmente cortadas em uma área de interesse de 60×70 *pixels*.

Buciu et al. apresentou em 2003 (BUCIU; KOTROPOULOS; PITAS, 2003) um sistema com taxa de acerto de 90,34% utilizando *Wavelet* de Gabor, SVM e validação com o *leave one out*. Novamente, as imagens foram manualmente recortadas e alinhas em uma região

de interesse de 160 × 120 *pixels* que, após uma sub-amostragem com fator 2, resulta em imagens de 80 × 60 *pixels* para extração de características.

Em 2004, Shinohara and Otsu (SHINOHARA; OTSU, 2004) propuseram um sistema com um extrator de características híbrido combinando *higher order local auto correlation* - HLAC e mapa de Fisher. Obtiveram uma taxa de 69,40% de acerto utilizando a validação cruzada. Os autores utilizaram seleção manual de uma área de interesse de face de 32 × 40 *pixels*, correção de iluminação e equalização de histograma.

Shih et al. (SHIH; CHUANG; WANG, 2008) alcançou uma taxa de acerto 94,13% utilizando o 2D-LDA como extrator de características, SMV com *kernel* polinomial e validação cruzada com 10-*fold*. Foi realizado um pré-processamento no banco através de recorte das imagens em áreas de face com resolução de 32 × 40 *pixels*, além de equalização de histograma.

A Tabela 5.6 apresenta o quadro comparativo com trabalhos de reconhecimento de expressão facial utilizando o Banco de Dados JAFFE.

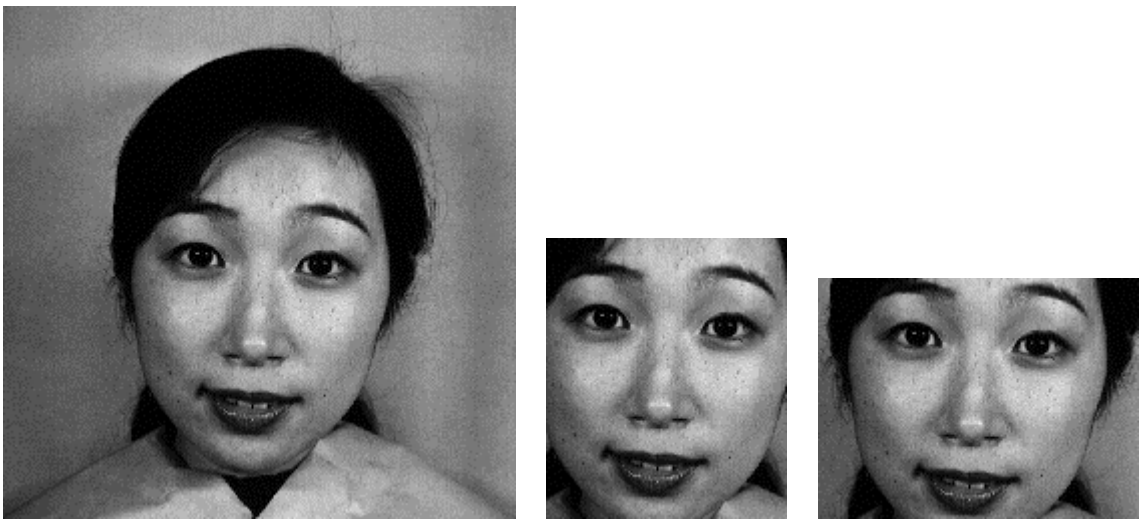
Método	**Validação**	**Taxa de Acerto [%]**
(ZHANG et al., 1998)	Cruzada	90,10
(LYONS; BUDYNEK; AKAMATSU, 1999)	Cruzada	92,00
(DUBUISSON; DAVOINE; MASSON, 2002)	Cruzada	87,60
(BUCIU; KOTROPOULOS; PITAS, 2003)	*Leave-One-Out*	90,34
(SHINOHARA; OTSU, 2004)	Cruzada	69,40
(SHIH; CHUANG; WANG, 2008)	Cruzada	94,13
Sistema Proposto (PEDROSO; SALLES, 2012)	*Leave-One-Out*	55,40

Tabela 5.6: Resultados do 3-NN para Face Neutra e Outra Expressão

Apesar de o sistema proposto apresentar a menor taxa de acerto com 55,40%, ressalta-se que a abordagem adotada não utiliza nenhum pré-processamento nas imagens de entrada do banco de dados e o reconhecimento é realizado de forma automática. Também salienta-se que a maneira como foi formulado e testado o sistema proposto introduz uma visão pessimista de classificação, pois os indivíduos testados na entrada do classificador sempre tinham todas as suas faces de todas as expressões simultaneamente de fora do processo de treinamento. Ou seja, o sistema deve reconhecer a expressão facial de um indivíduo nunca antes visto. O *leave one out* foi realizado para o indivíduo e não para uma imagem. O problema de reconhecer uma expressão facial de um indivíduo nunca apresentado ao banco possui maior complexidade do que identificar uma expressão de uma imagem não vista pelo banco, mas pertencente a um indivíduo já cadastrado. Isso sugere que implicitamente resultados de identificação dos indivíduos estejam inseridos no classificador. Portanto, uma expressão

facial de um novo indivíduo de entrada desconhecido pode ser classificada erroneamente como outra expressão pois o formato e textura da face apresenta forte correlação com outro indivíduo utilizado no treinamento.

Além disso, o sistema proposto nesse trabalho utilizou a imagem bruta, ou seja, sem translação, rotação, escalamento ou equalização de histograma. Essa imagem é entrada para o método de localização de faces Viola Jones, que é suscetível a falhas e/ou localização não exata do ponto de centro de face, pré-requisito para uma correta convergência do modelo sintético AAM para a imagem de entrada. Observe na Figura 5.24 que o comparativo entre a imagem original do banco de dados e as imagens recortadas utilizadas em outros trabalho indica que o pré-processamento de recorte de uma área de interesse é equivalente a substituição do bloco de detecção de faces no trabalho proposto. Observe ainda que no pré-processamento é eliminada a dificuldade de separar o objeto de interesse (face) do plano de fundo em regiões que apresentam níveis de cinza com intensidades em níveis semelhantes para as duas classes.



(a) (b) (c)

Figura 5.24: Comparativo entre a (a) imagem original utilizada no trabalho e imagens com pré-processamento de recorte de uma área de interesse de (b)120×140 *pixels* e (c) 160×120 *pixels*.

Capítulo 6

Conclusão

O reconhecimento de expressões faciais se desenvolve há mais de uma década e ainda é um tema recorrente e atual, bem como o reconhecimento de faces. Sua implementação utiliza embasamentos científicos descritos no campo da Filosofia e implementados com a Matemática, Engenharia e Computação.

As mais diversas aplicações e, em especial, as interfaces homem-máquina exigem sistemas eficientes, atuando no campo da computação pervasiva. Ou seja, é um sistema de base, invisível ao usuário comum, mas indispensável para as mais diversas tarefas.

Nesse trabalho dividiu-se o sistema de reconhecimento de expressões faciais em três etapas: localização da face, modelamento estatístico das expressões faciais e classificação.

Para a etapa de localização de faces foi utilizado o *framework* Viola-Jones, método amplamente difundido e aceito na literatura como um dos principais localizadores de face.

A modelagem estatística realizada através do *Active Appearance Model* - AAM apresentou bom poder representativo, robustez na extração de características e versatilidade em sua aplicação. Opera como bloco extrator de características.

Por fim, para o problema de classificação foi utilizado o método do vizinho mais próximo - NN e a da máquina de vetores de suporte - SVM.

6.1 Avaliação do Sistema

O JAFFE foi o Banco de Dados escolhido para treinamentos, testes e comparações com outros autores. O banco apresenta faces em tons de cinza com pouca variação de escala, translação, rotação e iluminação para as faces. Estão presentes as expressões faciais de alegria, tristeza, raiva, medo, surpresa, aversão e a face neutra.

Contribui-se com 68 *landmarks* marcados manualmente de acordo com o proposto no banco de dados CK+ e que podem ser utilizados em trabalhos de área correlata.

A abordagem proposta se mostrou efetiva na classificação das expressões faciais, uma vez que foi possível reconhecer as diferentes expressões faciais.

O módulo de detecção de face com o *framework* Viola-Jones localizou a totalidade das faces. Ele fornece um ponto estimado para o centro face. Foi proposto um sistema de distribuição de 9 pontos de testes ao redor do ponto fornecido pelo localizador de faces uma vez que a convergência do modelo estatístico é altamente sensível em relação ao ponto inicial de busca. O algoritmo de busca é iniciado em cada ponto e mantém-se como resultado o ponto que apresenta menor erro entre a imagem de entrada e o modelo sintético gerado.

O modelo AAM foi capaz de gerar um modelo estatístico de forma e textura para imagens que compõem um grupo de treino, além de sintetizar as novas imagens de entrada através de um algoritmo iterativo de busca. Como saída apresenta um vetor de aparência com dimensão reduzida de 65.536 referente aos 256×256 *pixels* para, no máximo, 26 características.

Para classificação foi utilizado em primeira instância um classificador estatístico K-NN. O primeiro teste avaliou a robustez do AAM e do K-NN utilizando um sistema de identificação para indivíduos com face neutra. 100% de acerto foi obtido, encorajando o acréscimo de novas expressões.

Para todos os testes seguintes foi utilizada a validação *leave one out* para cada indivíduo. Ou seja, a cada iteração de treino eram utilizadas as expressões de 9 indivíduos e as expressões do que ficou de fora são utilizadas como teste do sistema.

A validação cruzada apontou o menor erro para o caso de 3 vizinhos mais próximos sendo, portanto, adotado o 3-NN.

A medida que o número de classes aumenta a taxa de acerto diminui, indicando não lineari-

dade e correlação entre classes. Nos casos de dicotomia com a face neutra e outra expressão o sucesso ocorreu em 80,53% dos casos. Utilizando todas as expressões presentes no banco a taxa de acerto caiu para 47,42%.

Como alternativa para separar as classes não linearmente separáveis foi utilizado o classificador SVM com *kernel* RBF em uma tentativa de projetar os dados em um novo espaço de características onde as classes são melhor separáveis.

O parâmetro de custo C e o parâmetro de *kernel* γ foram calculados utilizando o 2-*fold* e o 5-*fold* para o grupo de treinamento. Para o caso envolvendo todas as classes o melhor desempenho foi obtido com o 2-*fold*. A taxa de acerto foi de 55,4% com sensibilidade 0,3667 e especificidade 0,9781, aumentando em 7,98% o percentual de acerto obtido anteriormente.

A Tabela 6.1 sumariza os resultados obtidos nos testes com o 3-NN e o SVM-RBF.

Expressões Utilizadas	**Taxa de Acerto [%] 3-NN**	**Taxa de Acerto [%] SVM - RBF 2-*fold***	**Taxa de Acerto [%] SVM - RBF 5-*fold***
N+R	86,67	73,33	81,67
N+Nj	79,96	83,05	67,80
N+M	70,97	85,48	74,19
N+F	80,33	80,33	90,16
N+T	85,25	60,66	88,52
N+S	80,00	65,00	85,00
N+F+T	73,91	52,17	88,04
N+F+T+S	65,57	41,80	72,13
N+F+T+M	62,10	53,23	82,26
N+F+T+M+Nj	41,18	51,63	65,36
N+F+T+S+M	57,79	41,56	61,04
N+R+F+T+S+M+N	47,42	55,40	45,07

Tabela 6.1: Resultados para classificação das expressões neutra (N), raiva (R), nojo (Nj), medo (M), felicidade (F), tristeza (T) e surpresa (S) utilizando K-NN

As expressões de tristeza e nojo foram as expressões com maiores problemas na modelagem AAM ao passo que todas as expressões neutras foram corretamente classificadas.

O indivíduo 7 apresentou falha em sua modelagem AAM para todas as suas expressões do banco e, excluindo-se esse caso extremo é possível atingir uma taxa de acerto de 60,10%. Esse resultado indica que o classificador é robusto e uma melhoria na implementação do algoritmo AAM acarretaria em maiores taxas de acerto.

Como contribuição maior destaca-se a proposta para mudança híbrida do vetor de pose no algoritmo de busca AAM utilizando como pesos uma parcela linear e outra não linear

relativa os autovalores obtidos no modelo combinado de treino. Essa mudança resultou em um menor erro entre o modelo sintético gerado pelo AAM e a imagem de entrada original em 12% dos casos.

6.2 Produção

A pesquisa desenvolvida nesse trabalho teve como fruto o artigo "Reconhecimento de Expressões Faciais baseada em Modelagem Estatística" publicado nos anais do CBA 2012 (PEDROSO; SALLES, 2012).

6.3 Trabalhos Futuros

Em trabalhos futuros será utilizado um banco de dados com um maior número de indivíduos e será incluída a expressão de desprezo (KANADE; COHN; TIAN, 2000). Além disso, o sistema de reconhecimento multi-classe pode ser expandido adicionando-se novas expressões faciais, até mesmo expressões não universais.

Um banco multi etnia também pode ser utilizado, haja visto que o treinamento com apenas uma etnia resulta em possíveis erros quando a imagem de entrada pertence a uma etnia nunca antes apresentada ao sistema.

Para a localização de face outras abordagens podem ser empregadas utilizando técnicas alternativas como, por exemplo, técnicas baseadas na detecção de pele, visando transpor as dificuldades inerentes ao processo como iluminação, baixo contraste, oclusão, escala, rotação e complexidade de plano de fundo. Além disso, pode-se acrescentar a possibilidade de detectar e extrair informação e modelo para múltiplas faces.

Outras características podem ser extraídas da face para o bloco de classificação. Com o *Facial Action Coding System*- FACS (EKMAN; FRIESEN, 1978), por exemplo, é possível mapear unidades de ação como o levantar de sobrancelhas ou abrir a boca. Portanto, a análise holística da imagem oferece resultados locais que podem ser entradas para o classificador.

Uma abordagem alternativa para o AAM é utilizar o LDA como redutor de dimensionalidade

e remapeamento das características ao invés do PCA, uma vez que Fisher proporciona uma maior separabilidade de classes, objetivo do sistema.

Outra informação que pode ser acrescentada no vetor de características é relativa ao tempo, sendo a análise baseada um vídeo, i.e. sequência de imagens. Portanto, pode-se aumentar a precisão do sistema. Além disso, a utilização de imagens coloridas pode aumentar a informação do modelo de textura.

Uma possível abordagem alternativa para classificação seria utilizar um treinamento AAM para cada expressão facial. No processo de decisão de classe seria atribuída à imagem de entrada o rotulo do modelo AAM que apresenta menor erro entre o modelo gerado e a imagem original.

Outra possibilidade é utilizar o AAM apresentado para estimativa de regiões da face como olhos, nariz e boco e, em seguida, utilizar o AAM para modelar essas pequenas áreas individualmente e com maior nível de informação.

Utilizar modelagem 3-D para o modelamento estatístico e incluir informação para o caso de oclusão também podem ser explorados.

No bloco de classificação a utilização de classificadores alternativos como o *Gaussian Mixture Model* e o HMM podem ser testados em casos de imagens em sequências de vídeos. Uma possível solução alternativa para elevar a taxa de acerto para o problema de multi-classe pode ser obtida utilizando uma combinação de classificadores.

Em outra aplicação, o reconhecimento de indivíduos tem objetivo dual do reconhecimento de faces: revelar a identidade de um protótipo independente de sua expressão facial. Uma vez detectada a expressão facial pode-se utilizar uma matriz de regressão com processo de treinamento similar ao descrito no algoritmo de busca AAM, mas apresentando exemplos de entrada de faces neutras. Portanto a busca é treinada para regredir de uma expressão específica para a neutra. Nesse caso, pode-se criar um bloco de pré-processamento capaz de gerar imagens com expressão neutra para um reconhecimento de face.

Por fim, o sistema pode ser empregado na geração de imagens sintéticas para formação de retratos falados. Dessa forma, a partir de uma única imagem podem ser geradas várias outras com diferentes expressões faciais.

6.4 Agradecimentos

Os autores agradecem a CAPES e ao PPGEE-UFES pelo incentivo, suporte e financiamento à pesquisa.

Também agradecem aos colaboradores que cederam imagens e banco de dados utilizados nos testes.

Apêndice A

Análise de Componentes Principais

A Análise de componentes principais ou, do inglês, *Principal Components Analysis*-PCA, é uma técnica utilizada para representar dados em uma nova base onde é possível obter uma redução dimensional armazenando a maior parte da informação em algumas componentes (BISHOP, 2006). Como consequência, pode-se obter redução do custo computacional em atividades como a classificação de dados.

O objetivo do PCA é mapear vetores $\mathbf{x}^n$, com $n = 1, 2, ..., N$ pertencentes a um espaço d-dimensional($\mathbf{x} = [x_1\ x_2\ ...\ x_d)$ em vetores $\mathbf{z}^n$ em um espaço M-dimensional ($\mathbf{z} = [z_1\ z_2\ ...\ z_d)$, onde $M < d$ e a variância dos dados é maximizada na nova base.

A.1 Fundamentação Teórica

Começamos a análise reescrevendo $\mathbf{x}$ como uma soma ponderada de d vetores ortonormais em um novo espaço denotados por $\boldsymbol{u_i}$:

$$\mathbf{x} = \sum_{i=1}^{d} z_i \boldsymbol{u_i}, \tag{A.1}$$

onde

$$\boldsymbol{u_i}^T \boldsymbol{u_i} = \delta_{ij}. \tag{A.2}$$

A ortonormalidade é garantida pelo delta de *Kronecker* δ_{ij} definido como

$$\delta_{ij} = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j. \end{cases} \tag{A.3}$$

Portanto, os valores para z_i podem ser calculados conforme

$$z_i = \boldsymbol{u_i}^T \mathbf{x}. \tag{A.4}$$

Observa-se que a nova representação pode ser vista como uma rotação nas coordenadas do sistema original. Uma redução de dimensionalidade pode ser obtida aproximando $\mathbf{x}$ por $\tilde{\mathbf{x}}$ se escolhermos um subconjunto de $M < d$ vetores de base $\boldsymbol{u_i}$

$$\tilde{\mathbf{x}} = \sum_{i=1}^{M} z_i \boldsymbol{u_i} + \sum_{i=M+1}^{d} b_i \boldsymbol{u_i}, \tag{A.5}$$

onde $\boldsymbol{b_i}$ é a o conjunto de vetores de base correspondentes aos z_i não utilizados. A redução de dimensionalidade deve ocorrer minimizando o erro devido à aproximação da função

$$\boldsymbol{x_n} - \tilde{\mathbf{x}} = \sum_{i=M+1}^{d} \left(z_n^i - b_i \right) \boldsymbol{u_i}. \tag{A.6}$$

Se consideramos o método dos mínimos quadrados, o erro E_M é definido como

$$E_M = \frac{1}{2} \sum_{n=1}^{M} \|\mathbf{x}^n - \tilde{\boldsymbol{x}}^{\boldsymbol{n}}\| = \frac{1}{2} \sum_{n=1}^{M} \sum_{M+1}^{d} (z_n^i - b_i)^2. \tag{A.7}$$

A minimização do erro pode ser obtida fazendo $\frac{\partial E_M}{\partial b_i}$. Nesse caso

$$b_i = \frac{1}{N}\sum_{n=1}^{N} z_i^n = \boldsymbol{u_i}\bar{\mathbf{x}}, \tag{A.8}$$

onde $\bar{\mathbf{x}}$ é o vetor de média para os dados originais calculados conforme

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}^n. \tag{A.9}$$

Utilizando os resultados das Equações A.4 e A.9, o erro E_M pode ser reescrito como

$$E_M = \frac{1}{2}\sum_{i=1}^{M}\sum_{M+1}^{d} [\mathbf{u}_i^T(\mathbf{x}^n - \bar{\mathbf{x}})] \tag{A.10}$$

$$E_M = \frac{1}{2}\sum_{M+1}^{d} \mathbf{u}_i^T \boldsymbol{\Sigma} \mathbf{u}_i, \tag{A.11}$$

onde $\boldsymbol{\Sigma}$ é a matriz de covariância dos dados dada pela Equação A.12.

$$\boldsymbol{\Sigma} = \sum_{n} (\mathbf{x}^n - \bar{\mathbf{x}})\,(\mathbf{x}^n - \bar{\mathbf{x}})^T \tag{A.12}$$

Por fim, deve-se escolher quais vetores $\mathbf{u}_i$ formarão a base do novo espaço. A minimização do erro E_M ocorre quando os vetores de base satisfazem a equação

$$\Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i, \tag{A.13}$$

onde λ_i são os autovalores associados ao autovetores $\mathbf{u}_i$ que compõem a matriz de covariância Σ (DUDA; HART; STORK, 2001). Logo, o erro de aproximação pode ser reescrito em função dos autovalores λ_i, conforme

$$E_M = \frac{1}{2}\sum_{i=M+1}^{d} \lambda_i. \tag{A.14}$$

Logo, o menor erro está associado à escolha dos $d - M$ menores autovalores λ_i. Observe que quando todos os autovalores são mantidos, ou seja, todos $\mathbf{u}_i$ formam a base do novo espaço, nenhuma redução de dimensionalidade é obtida e o erro é igual a zero. Além disso, esse resultado mostra que as componentes mais significativas, ou seja, aquelas com maiores variância dos dados, estão associadas aos maiores autovalores.

A.2 Resultados Experimentais

O algoritmo PCA foi testado em dados 2-D e, posteriormente, em um banco com um maior número de características.

A.2.1 Dados Bidimensionais

Como teste preliminar, foi criado uma matriz de dados utilizando 2 características para um conjunto de 100 dados, ou seja, uma matriz 2x100. Em cada direção foi atribuída uma variância diferente para os dados utilizando geradores aleatórios com distribuição normal e desvio padrão de 5 e 2 em cada um de seus eixos. Em seguida, foi realizada uma rotação de 45 ° através de uma matriz de rotação para que a maior variância não ocorresse na base original cartesiana. Através da análise dos autovalores e autovetores da matriz de covariância foi possível gerar uma nova base que maximiza a variância dos dados quando projetados nela. Os resultados são exibidos na Figura A.1

A Figura A.2.1 ilustra a projeção dos dados em cada um dos eixos da nova base. Observa-se, nesse caso, que os autovalores $\lambda_1 = 20,0074$ e $\lambda_2 = 4,1651$ correspondem aos autovetores $\mathbf{u}_1 = [-0.7376\ 0.6752]$ e $\mathbf{u}_2 = [-0.6752\ -0.7376]$, respectivamente. Logo, a maior variância dos dados ocorrem em $\mathbf{u}_1$.

A.2.2 Dados Tridimensionais

O problema bidimensional foi estendido para um problema tridimensional criando um banco de dados com três características com distribuição normal e de desvio padrão de 0,2, 4 e 5 nas direções $\mathbf{x}_1$, $\mathbf{x}_2$ e$\mathbf{x}_3$, respectivamente. Além disso, os dados estão distribuídos em uma rotação de 45 ° em relação ao eixo cartesiano. A matriz de covariância é levantada segundo

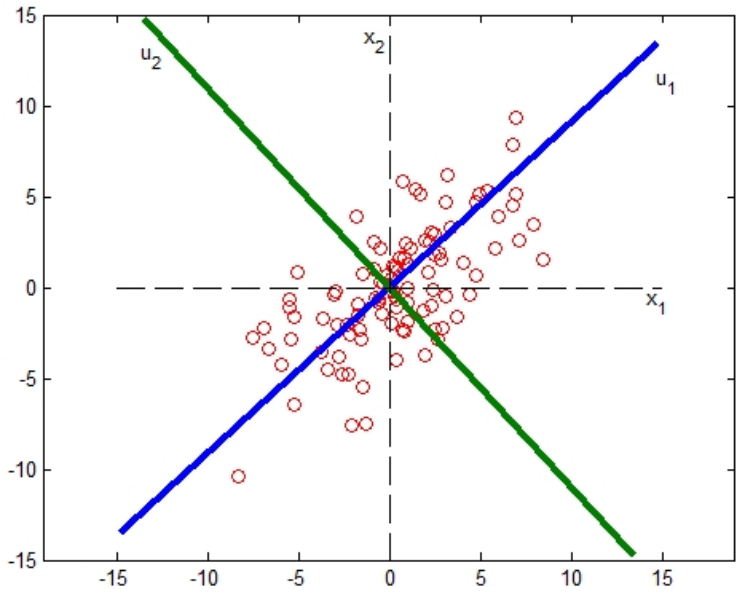
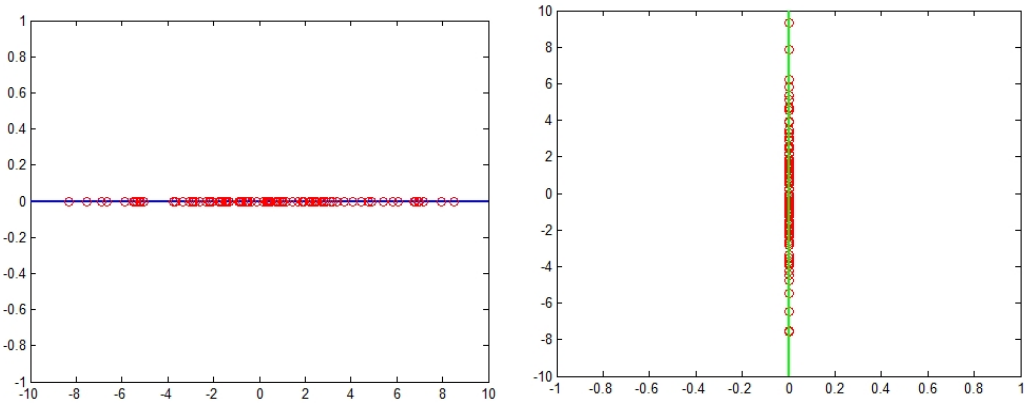


Figura A.1: Os dados apresentando na base original formada por $\mathbf{x}_1$ e $\mathbf{x}_2$ podem ser representandos por uma nova base formada por $\mathbf{u}_1$ e $\mathbf{u}_2$, onde é maximizada a variância dos dados. Observe que a maior parte da informação pode ser obtida projetando $\mathbf{x}$ em $\mathbf{u}_1$.



(a) (b)

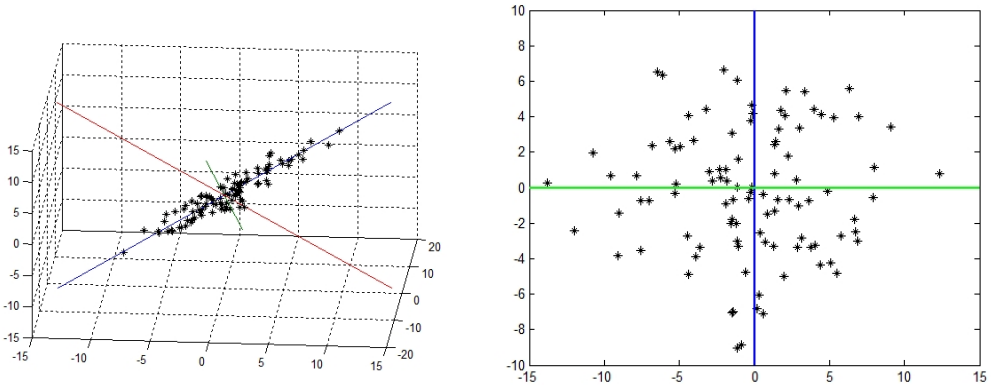
Figura A.2: Dados projetados em cada uma das componentes da nova base. Em (a) os dados são projetados em $\mathbf{u}_1$ e tem-se uma maior variância associada ao maior autovalor da matriz de covariância. Em (b) tem-se a projeção em $\mathbf{u}_2$.

Equação A.12 e os autovalores e autovetores associados obtidos estão sumarizados na Tabela A.1.

A Figura A.3(a) ilustra os dados projetados nos eixos cartesianos e são indicadas as bases que maximizam a variância dos dados. Observe que ao projetarmos os dados nas duas principais componentes é obtida uma redução de dimensão, conforme Figura A.7(b).

Autovalor	**Autovetor Associado**
$\lambda_1 = 23,0294$	$\mathbf{u}_1 = [-0,6913 \ -0,1420 \ -0,7085]$
$\lambda_2 = 12,8438$	$\mathbf{u}_2 = [-0,2073 \ 0,9783 \ 0,0062]$
$\lambda_3 = 0,0317$	$\mathbf{u}_3 = [-0,6922 \ -0,1512 \ 0,7057]$

Tabela A.1: Autovalores e autovetores associados para os dados tridimensionais.



(a) (b)

Figura A.3: (a) Dados tridimensionais e as bases do espaço que maximiza a variância. (b) Projeção dos dados nas duas componentes principais.

A.2.3 Testes com o Banco de Dados Iris

O teste final foi tratado em um problema de classificação utilizando o banco de dados Iris (FISHER, 1936). Esse *database* armazena os dados referentes a 150 plantas contendo quatro características: comprimento e largura da sépala e comprimento e largura da pétala. Portanto, temos os dados distribuídos em uma matriz 4x150. Além disso, as primeiras 50 amostras são da espécie setosa, as 50 seguintes da versicolor e as 50 últimas da virginica. Os dados são exibidos na Figura A.4.

Observa-se claramente que o comprimento de pétala é o dado mais fácil de ser separado, uma vez é possível estabelecer um limiar inferior e um limiar superior capazes de separar essa característica para as três classes. Portanto, para fins de observar uma plotagem em três dimensões, serão exibidos os resultados para o comprimento da sépala, largura da sépala e largura da pétala. O dado de comprimento de pétala será suprimido na exibição dos resultados, ainda que existam quatro dimensões. Esses dados são apresentados na Figura A.5.

A Figura A.6 apresenta os dados mapeados na nova base e sem efeito da média.

Na Figura A.7 são exibidos os dados projetados em subespaços de dimensão dois formados

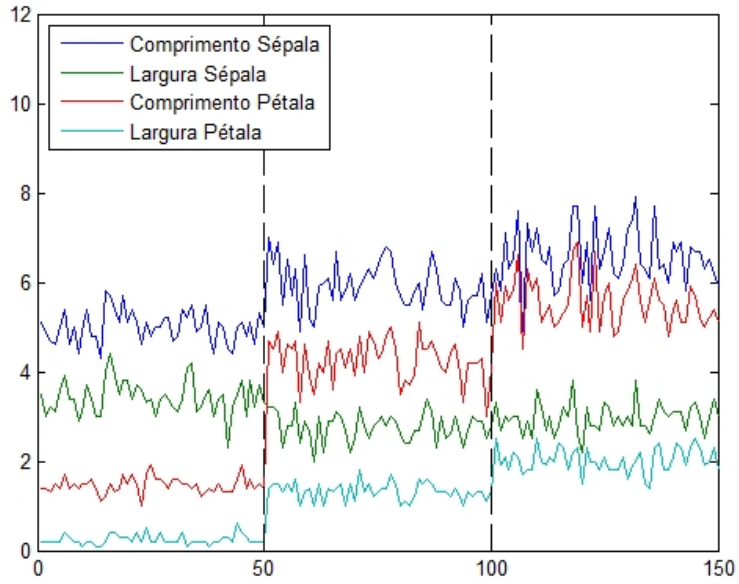


Figura A.4: Distribuição das quatro características do banco de dados Iris (FISHER, 1936): comprimento e largura de sépala e pétala.

por combinações dois a dois das componentes $\mathbf{u}_1$, $\mathbf{u}_2$ e $\mathbf{u}_3$. Observa-se que com apenas as informações presentes nos dados projetados no subespaço $\mathbf{u}_1$ e $\mathbf{u}_2$ ou $\mathbf{u}_1$ e $\mathbf{u}_3$ é possível separar a classe Setosa das demais. Isso evidencia como é forte a variância no eixo $\mathbf{u}_1$ em relação a $\mathbf{u}_2$ e $\mathbf{u}_3$.

A Tabela A.2 sumariza os autovalores e seus respectivos autovetores associados a matriz de covariância dos dados em seu espaço original.

Autovalor	**Autovetor Associado**
$\lambda_1 = 4,2282$	$\mathbf{u}_1 = [0,3614\ 0,6566\ 0,5820\ -0,3155]$
$\lambda_2 = 0,2427$	$\mathbf{u}_2 = [-0,0845\ 0,7302\ -0,5979\ 0,3197]$
$\lambda_3 = 0,0782$	$\mathbf{u}_3 = [0,8567\ -0,1734\ -0,0762\ 0,4798]$
$\lambda_4 = 0,0238$	$\mathbf{u}_3 = [0,3583\ -0,0755\ -0,5458\ -0,7537]$

Tabela A.2: Autovalores e autovetores associados para os dados tridimensionais.

Quanto maior o número de componentes utilizadas no espaço de características menor será o erro, ou seja, a informação é mantida com maior precisão na nova base. A Figura A.8 relaciona o erro para cada uma das características (comprimento e largura de pétala e sépala) utilizando um número variado de componentes. Com todos os autovetores associados a todos os autovalores o erro é nulo. A medida que o número de componentes principais é reduzida o erro aumenta.

Esse resultado reforça que com a redução de dimensionalidade é possível representar um

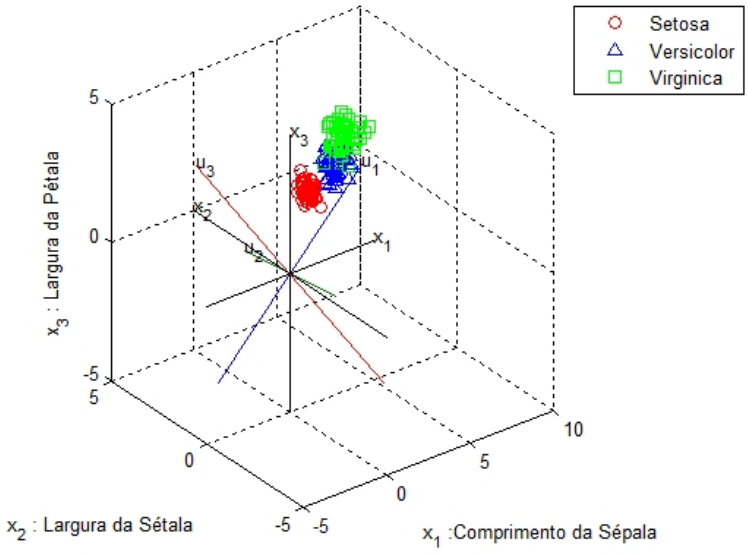


Figura A.5: Distribuição das três classes e suas três principais características. A espécie setosa é linearmente separável das outras duas classes.

dado com perda de informação controlada de modo que a relação custo computacional e complexidade computacional x precisão é favorável a etapas futuras de processamento como, por exemplo, a técnica de classificação baseada na máquina de vetores de suporte utilizando os dados em uma nova base reduzida.

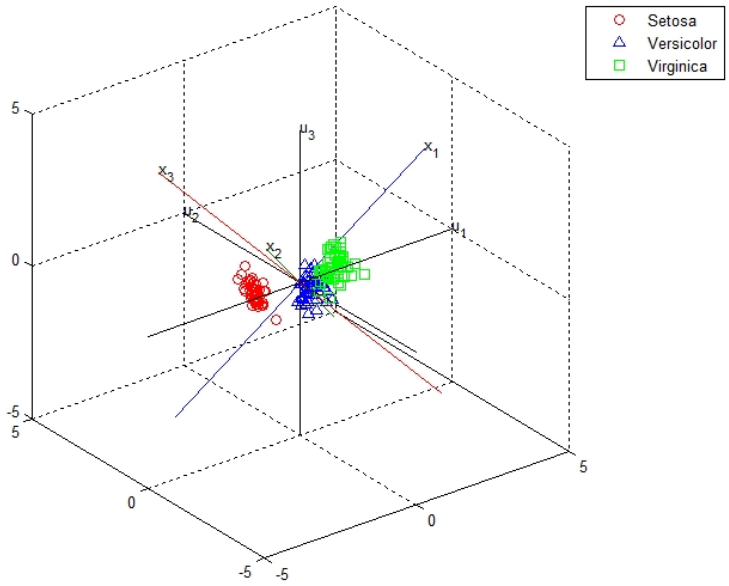


Figura A.6: Mapeamento dos dados na nova base sem efeito da média de cada característica.

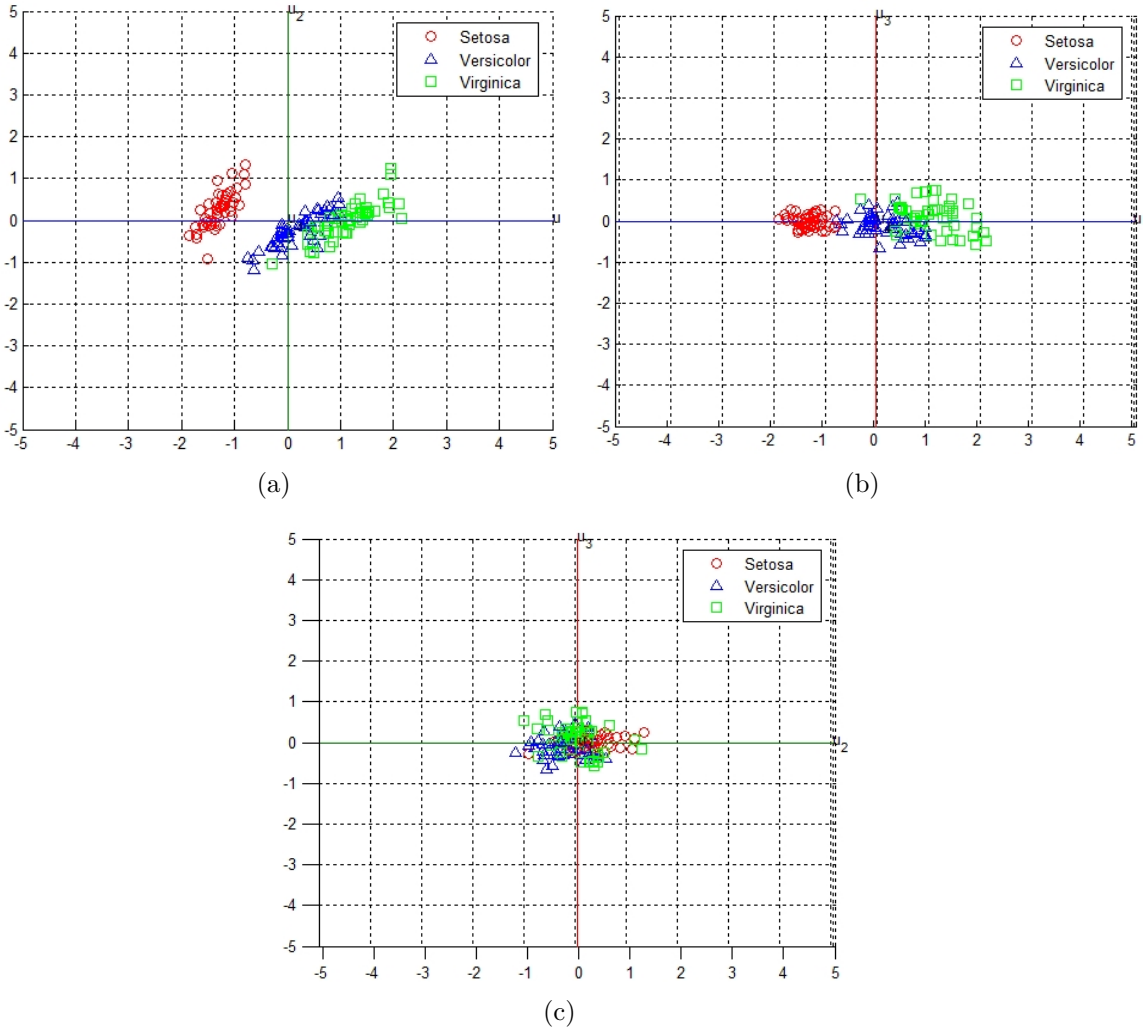


Figura A.7: Projeção dos dados nas novas bases considerando como base (a)$\mathbf{u}_1$ e $\mathbf{u}_2$; (b) $\mathbf{u}_1$ e $\mathbf{u}_3$; (c) $\mathbf{u}_2$ e $\mathbf{u}_3$.

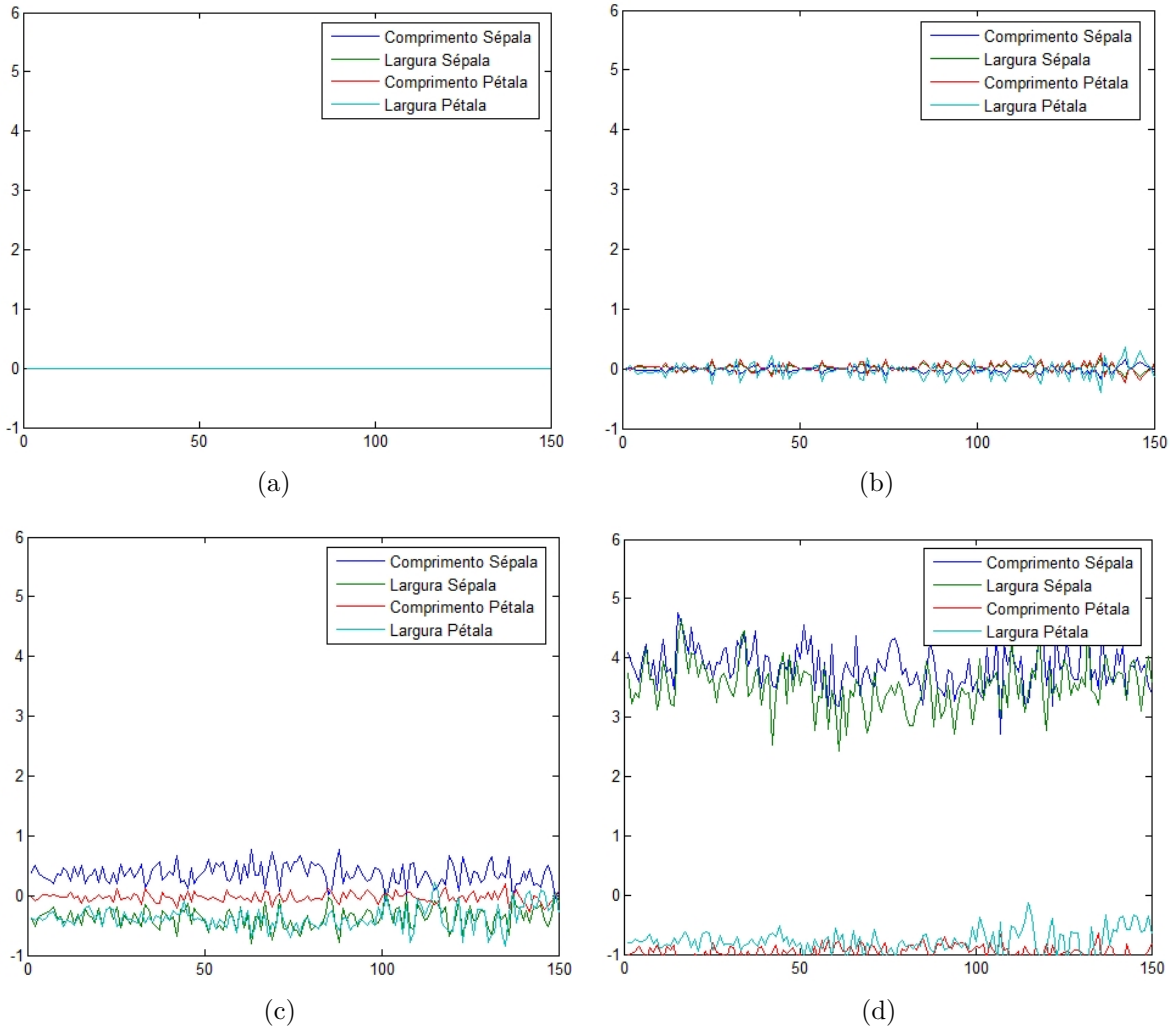


Figura A.8: Erro $(E_M)^2$ obtido utilizando: (a) 4 Componentes Principais; (b) 3 Componentes Principais; (c) 2 Componentes Principais (d) 1 Componente Principal.

Apêndice B

Adaboost

A melhora adaptativa, *Adaptative Boosting* ou, simplesmente, *Adaboost*, é um algoritmo apresentado em 1995 por Freund e Schapire (FREUND; SCHAPIRE, 1997) que tem como objetivo melhorar a precisão de um algoritmo de aprendizagem.

Sua estratégia é criar uma série de classificadores a partir de um banco de dados e atribuir pesos para cada regra de decisão. Ou seja, a decisão final é baseada na ponderação de vários classificadores. Com isso, espera-se atribuir um peso maior a determinados exemplos do banco de dados que sejam mais representativos para certas situações. Por exemplo, realizando-se o levantamento das notas de alunos em uma sala de aula observa-se uma flutuação das notas individuais. No entanto, espera-se que o aluno com maior notas nas últimas avaliações também seja bem avaliado, ainda que tenha um histórico negativo em avaliações passadas.

Consideraremos o caso de um algoritmo que deve separa duas classes ω_1 e ω_2. Para os m dados de treino $\boldsymbol{x}_i \in X$, onde $i = 1, 2, ..., m$, existe associado um vetor de alvo $y_i \in Y = \{-1 \ +1\}$ que determina a qual das duas classes pertence a amostra.

Forma-se o conjunto de dados $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_j, y_j), ..., (\boldsymbol{x}_m, y_m)$.

O algoritmo de aprendizagem a ser otimizado é chamado de algoritmo de aprendizagem fraco ou algoritmo de aprendizagem base. No caso do *framework* de detectção de face Viola-Jones (VIOLA; JONES, 2001) é utilizado a rede neural *perceptron*. O algoritmo base é utilizado em uma série de T repetições. Para cada repetição, cada exemplo de treino $\boldsymbol{x}_i$ existe associado um peso $D_t(i)$ para cada uma das $t = 1, 2, ..., T$ repetições. Inicialmente todos os pesos são

iguais (distribuição uniforme). A cada iteração observa-se quais dados $\mathbf{x}_i$ são classificados erroneamente e, na próxima iteração, aumenta-se o peso $D_t(i)$ associado aos exemplos que apresentaram erro. Dessa forma, a cada iteração o algoritmo de aprendizagem fraco deve aumentar o foco nos exemplos problemáticos.

O classificador fraco deve formular uma hipótese fraca h_t associada a $D_t(i)$ para rotular os dados de treino, fazendo

$$h_t \ : \ X \rightarrow \{-1 \ +1\}. \tag{B.1}$$

A eficiência da hipótese de classificação fraca é medida através do erro ϵ_t definido como

$$\epsilon_t = Pr_{i\sim D_t}[h_t(\mathbf{x}_i \neq y_i)] = \sum_{i \ : \ h_t(\mathbf{x}_i)\neq y_i)} D_t(i). \tag{B.2}$$

Em seguida deve-se atualizar a distribuição de $D_t(i)$ ou, equivalentemente, atualizar o peso atribuído a cada exemplo de treino para efeito de classificação baseado no erro ϵ_t. Atrelado ao erro tem-se uma variável α_t responsável por medir a importância atribuída a hipótese h_t calculada segundo

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right). \tag{B.3}$$

Observe que a medida que aumenta-se α_t diminui-se ϵ_t. Além disso, $\epsilon_t \leq \frac{1}{2}$ se $\alpha_t \geq 0$.

A atualização dos pesos $D_t(i)$ é realizada através das relações

$$D_{t+1} = \begin{cases} \frac{D_t(i)e^{-\alpha_t}}{Z_t}, & \text{se } h_t(i) = y_i \\ \frac{D_t(i)e^{+\alpha_t}}{Z_t}, & \text{se } h_t(i) \neq y_i \end{cases} \tag{B.4}$$

e

$$\frac{D_t(i)e^{-\alpha_t y_i h_t(i)}}{Z_t}, \tag{B.5}$$

onde Z_t é o fator de normalização para D_t ser uma distribuição ($\sum_{i=1}^{m} D_t(\mathbf{x}_i) = 1$).

O processo de atualização dos pesos ocorre até a última iteração ser atingida, ou seja, $t = T$. Finalmente, um classificador forte (saída do *Adaboost* é determinado como a ponderação dos resultados de todas as iterações. Para uma entrada $\mathbf{x}$, a hipótese final H é dada por (FREUND; SCHAPIRE; ABE, 1999):

$$H(\mathbf{x}) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right). \tag{B.6}$$

Apêndice C

Mapa de Sammon

Os sentidos dos seres humanos estão acostumados a trabalhar com dados de 2 ou 3 dimensões. No entanto, na área de Processamento de Sinais comumente nos deparamos com dados em maiores dimensões. Nessas situações os processo de análise de dados e separabilidade de classes tornam-se difíceis e até mesmo inviáveis.

Em 1969, Jhon W. Sammon propôs um algoritmo de mapeamento não linear capaz de mapear em um novo espaço vetores pertencentes a um espaço de maior ordem preservando a estrutura dos dados no que tange o aspecto de relações geométricas entre os vetores originais (SAMMON, 1969). Portanto, é realizada uma redução de dimensionalidade e no novo espaço tenta-se preservar as distâncias entre os vetores existente no espaço original.

Há um particular interesse no mapeamento dos dados em 2 e 3 dimensões para análise visual dos seres humanos.

C.1 Fundamentação Teórica

Deseja-se utilizar o Mapa de Sammon para realizar a projeção de dados segundo $\Re^L \rightarrow \Re^d$.

Defini-se um conjunto de N vetores $\mathbf{X}_i$ pertencentes a um espaço L-dimensional, onde $i = [1, 2, ..., N]$ e $\mathbf{X}_i = [x_i1\ x_i2\ ...\ x_{iL}]$.

De maneira similar defini-se um conjunto de N vetores $\mathbf{Y}_i$ pertencentes a um espaço d-

dimensional, onde $i = [1, 2, ..., d]$ e $\mathbf{X}_i = [x_i1\ x_i2\ ...\ x_{id}]$. O vetor $\mathbf{Y}_i$ deve ser a representação de $\mathbf{X}_i$ no espaço de menor dimensão e, em especial, $d = 2$ ou $d = 3$.

A distância entre duas entradas $\mathbf{X}_i$ e $\mathbf{X}_j$ no espaço original é tomada como

$$d^*_{i,j} \equiv dist\left[\mathbf{X}_i, \mathbf{X}_j\right], \tag{C.1}$$

onde a distância *dist* representa alguma medida geométrica como, por exemplo, a Distância Euclidiana.

A distância no espaço de menor ordem é dada por

$$d_{i,j} \equiv dist\left[\mathbf{Y}_i, \mathbf{Y}_j\right]. \tag{C.2}$$

Como a princípio a localização dos pontos originais não é conhecida, atribui-se aleatoriamente valores iniciais para $\mathbf{Y}_i$.

Em seguida, calcula-se a distância entre os dados no espaço L-dimensional original e os dados remapeados no espaço d-dimensional. É verificado se as posições dos dados projetadas no espaço de menor dimensão condizem com as relações presentes no espaço original através do Erro de Sammon ou Erro ou *Stress* E definido como

$$E = \frac{1}{C}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left(\frac{\left[d^*_{i,j} - d_{i,j}\right]^2}{d^*_{i,j}}\right), \tag{C.3}$$

onde C é a distância total entre os dados no espaço original dada por

$$C = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left[d^*_{i,j}\right]. \tag{C.4}$$

A Equação C.3 do Erro E pode ser sintetizada segundo

$$E = \frac{1}{\sum\limits_{i<j}\left[d^*_{i,j}\right]} \sum_{i<j}^{N} \left(\frac{\left[d^*_{i,j} - d_{i,j}\right]^2}{d^*_{i,j}} \right). \tag{C.5}$$

Portanto, observe que o erro é dependente de $d \times N$ variáveis associadas a cada vetor projetado no espaço d-dimensional. Associado a cada vetor tem-se um termo $y_{p,q} \equiv y_{pq}$, onde $p = [1, 2, ..., N]$ e $q = [1, 2, ..., d]$ refletem no valor do erro total E.

Uma vez calculado o erro, espera-se diminuí-lo, isto é aproximar o mais fielmente possível as relações dos dados remapeados em relação ao espaço original. Isso pode ser realizado em um procedimento iterativo realocando os vetores $\mathbf{Y}_i$ de acordo com o erro obtido. Uma solução é a utilização da Descida de Gradiente, onde os vetores $\mathbf{Y}_i$ devem ser deslocados da direção da maior variação do gradiente do erro para localização de um mínimo local. Equivalentemente, deve-se ajustar y_{pq}, conforme justificado a seguir.

Considere o erro $E(m)$ associado a m-ésima iteração como

$$E \equiv \frac{1}{C} \sum_{i<j}^{N} \left(\frac{\left[d^*_{i,j} - d_{i,j}(m)\right]^2}{d^*_{i,j}} \right), \tag{C.6}$$

onde

$$d_{i,j}(m) = \sqrt{\sum_{k=1}^{d} [y_{ik}(m) - y_{jk}(m)]}. \tag{C.7}$$

A configuração dos vetores projetados pelo remapeamento de Sammon no novo espaço é alterada na iteração $m + 1$ seguindo a atualização dos vetores conforme

$$y_{pq}(m + 1) = y_{pq}(m) - \alpha \Delta_{pq}(m), \tag{C.8}$$

onde $\alpha \approx 0.3$ ou $\alpha \approx 0.4$ é uma constante definida empiricamente para convergência do algoritmo através de experimentos de Sammon e

$$\Delta_{pq}(m) = \frac{\partial E(m)}{\partial y_{pq}(m)} \Big/ \left| \frac{\partial E^2(m)}{\partial y_{pq}^2(m)} \right| \tag{C.9}$$

As derivadas parciais são dadas pelas equações

$$\frac{\partial E(m)}{\partial y_{pq}(m)} = \frac{-2}{C} \sum_{\substack{j=1 \\ j \neq p}}^{N} \left[\frac{d_{pj}^* - d_{pj}}{d_{pj}^* d_{pj}} \right] (y_{pq} - y_{jq}) \tag{C.10}$$

e

$$\frac{\partial E^2(m)}{\partial y_{pq}^2(m)} = \frac{-2}{C} \sum_{\substack{j=1 \\ j \neq p}}^{N} \frac{1}{d_{pj}^* d_{pj}} \left[\left(d_{pj}^* - d_{pj} \right) - \frac{(y_{pq} - y_{jq})}{d_{pj}} \left(1 + \frac{d_{pj}^* - d_{pj}}{d_{pj}} \right) \right] \tag{C.11}$$

Ressalta-se que devido à formulação matemática do problema é necessária na etapa de pré-processamento garantir que não existem dados sobreposto na entrada, pois isso acarreta estouro no cálculo das derivadas parciais.

O critério de parada do algoritmo pode ser baseado no Erro de Sammon ou em um número máximo de iterações.

O algoritmo apresenta dois pontos a serem analisados. O primeiro diz respeito a utilização da Descida de Gradiente, já que o erro minimizado é um mínimo local podendo ser diferente do mínimo global. Outro ponto é o custo computacional estimado como $O(N^2)$ já que são necessários $\frac{N \times (N-1)}{2}$ cálculos de distância para os N pontos. Uma solução para otimização do algoritmo é utilizar a Análise de Componentes Principais - PCA (LERNER et al., 2000) para estimar a projeção dos pontos no espaço $\Re^d$ e utilizá-la como estimativa inicial de $\mathbf{Y}_i$.

C.1.1 Testes com o Banco de Dados Iris

O teste do Mapeamento de Sammon foi realizado utilizando o banco de dados Iris (FISHER, 1936), também utilizado e apresentado no Apêndice A. O banco de dados é formado por 50 plantas de cada uma das três espécies: setosa,versicolor e virginica. Os dados são armazenados em um espaço 4-dimensional ($L = 4$) em que o vetor de dados $\mathbf{X}_i$ é formado pelas características de comprimento e largura da sépala e comprimento e largura da pétala.

A Análise de Componentes Principais sugere que a espécie setosa é linearmente separável das outras duas classes, conforme observa-se na Figura A.5.

Utilizando a projeção bidimensional dos dados no Mapa de Sammon confirmamos esse resultado, como ilustra a Figura C.1. Nesse caso, o mapeamento de Sammon é capaz de reduzir o espaço original 4-D para um espaço 2-D preservando as relações de distância originais dos vetores.

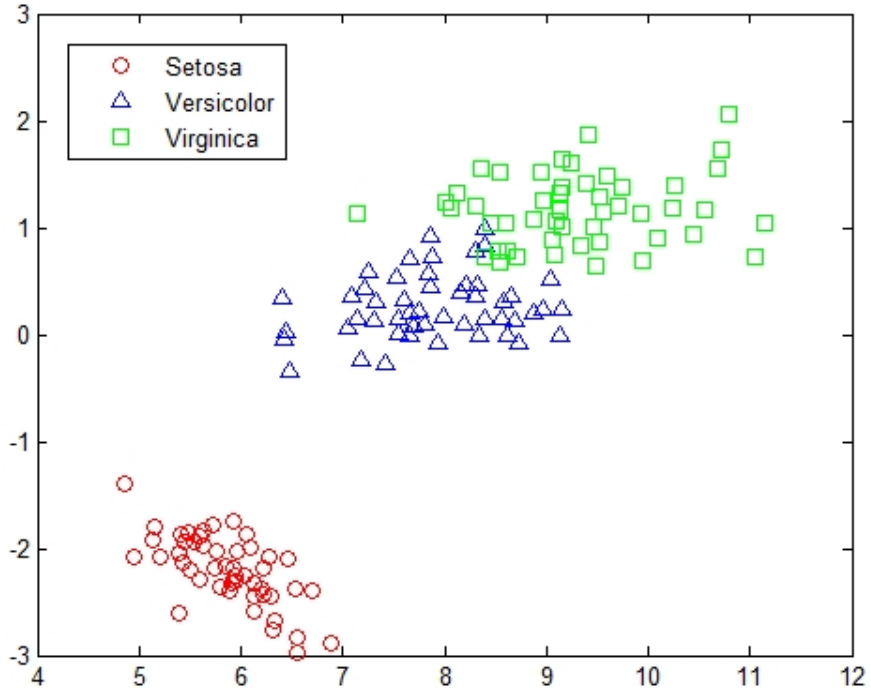
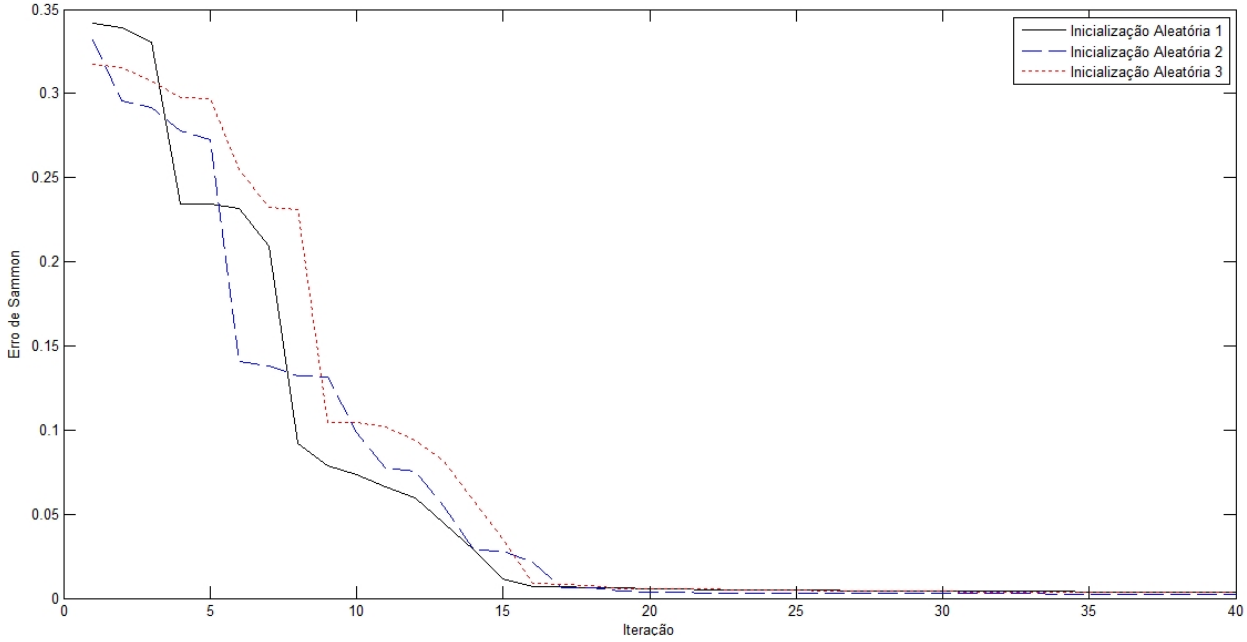
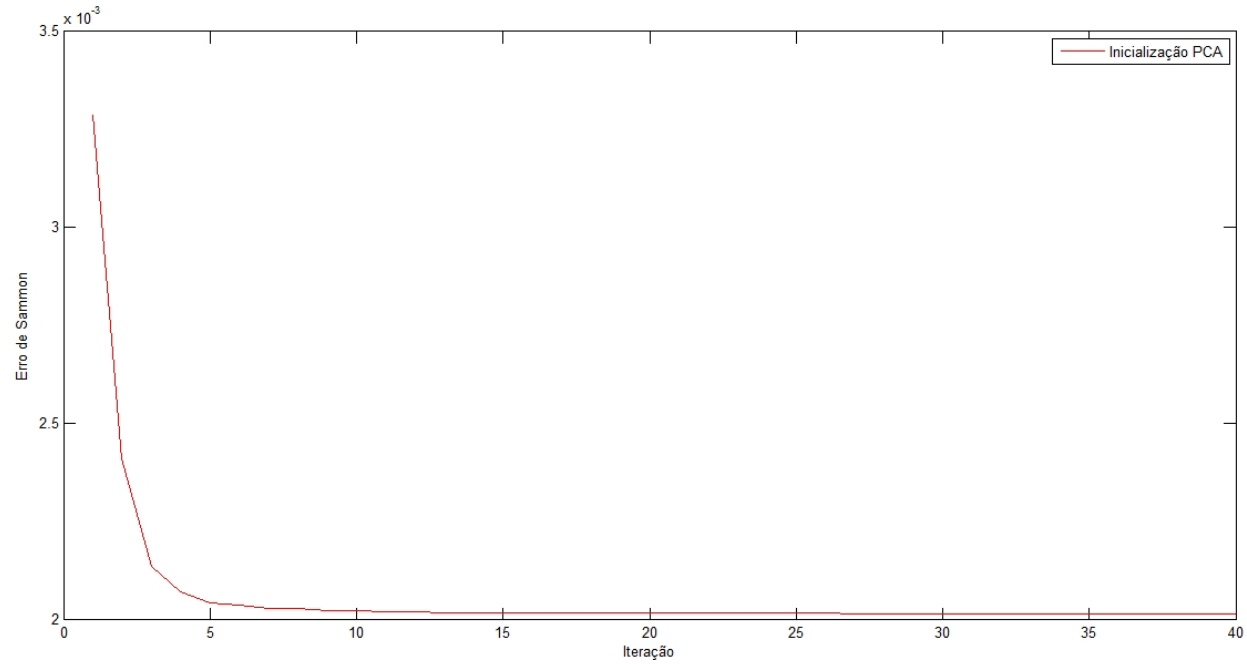


Figura C.1: Redução do espaço original 4-D para um espaço 2-D preservando as relações de distância originais dos vetores através do Mapa de Sammon.

A convergência do algoritmo em função do Erro de Sammon para o banco de dados Iris é apresentada na Figura C.2. Observa-se uma rápida convergência da inicialização utilizando o PCA enquanto a inicialização aleatória apresenta um maior número de iterações necessárias para convergência, além de maiores oscilações no erro, incluindo casos onde há acréscimo do erro entre iterações sucessivas.



(a)



(b)

Figura C.2: Comparação do Erro de Sammon utilizando inicialização (a) aleatória para os pontos projetados na primeira iteração de Sammon e (b) inicialização utilizando PCA.

Referências Bibliográficas

AHMED, F.; KABIR, M. Directional ternary pattern (dtp) for facial expression recognition. In: *Consumer Electronics (ICCE), 2012 IEEE International Conference on.* [S.l.: s.n.], 2012. v. 2, p. 265 –266. ISSN 2158-3994.

BARTLETT, M. S. et al. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In: *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on.* [S.l.: s.n.], 2003. v. 5, p. 53. ISSN 1063-6919.

BETTADAPURA, V. Face expression recognition and analysis: The state of the art. *ArXiv e-prints*, p. 1–27, março 2012.

BISHOP, C. M. *Pattern Recognition and Machine Learning.* [S.l.]: Springer, 2006. 738 p. (Information science and statistics, 4). ISSN 10179909. ISBN 9780387310732.

BUCIU, I.; KOTROPOULOS, C.; PITAS, I. ICA and Gabor representation for facial expression recognition. In: *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on.* [S.l.: s.n.], 2003. v. 2, p. II – 855–8 vol.3. ISSN 1522-4880.

CALABRESE, F. et al. Workshop on pervasive urban applications. *Pervasive Computing, IEEE*, v. 10, n. 4, p. 101 –104, abril 2011. ISSN 1536-1268.

CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, ACM, New York, NY, USA, v. 2, n. 3, p. 27:1–27:27, 2011. ISSN 2157-6904.

CHANG, C.-Y.; HUANG, Y.-C. Personalized facial expression recognition in indoor environments. In: *Neural Networks (IJCNN), The 2010 International Joint Conference on.* [S.l.: s.n.], 2010. p. 1 –8. ISSN 1098-7576.

CHIBELUSHI, C. C.; BOUREL, F. Facial expression recognition: A brief tutorial overview. *On-Line Compendium of Computer Vision*, 2003.

COHEN, I. et al. Learning bayesian network classifiers for facial expression recognition both labeled and unlabeled data. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on.* [S.l.: s.n.], 2003. v. 1, p. I–595 – I–601 vol.1. ISSN 1063-6919.

COOTES, T.; EDWARDS, G. J.; TAYLOR, C. Active appearance models. *Proceedings of the 5th European Conference on Computer Vision*, v. 2, p. 484–498, 1998.

COOTES, T. F.; TAYLOR, C. J. *Statistical Models of Appearance for Computer Vision*. 2004. World Wide Web Publication February. `http://www.isbe.man.ac.uk/~bim/Models/app_models.pdf`. Acessado em abril de 2012.

CORTES, C.; VAPNIK, V. Support-Vector Networks. *Machine Learning*, Kluwer Academic Publishers, v. 20, n. 3, p. 273–297, 1995. ISSN 0885-6125.

COURANT, R.; HILBERT, D. *Methods of Mathematical Physics*. [S.l.]: Wiley-Interscience, 1953. 560 p.

COVER, T.; HART, P. B. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IEEE, v. 13, n. 1, p. 21–27, 1967. ISSN 00189448.

DARWIN, C. *The Expression of the Emotions in Man and Animals*. London: John Murray, 1872.

DONATO, G. et al. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, MIT Press, v. 21, n. 10, p. 974–989, 1999. ISSN 19393539.

DUBUISSON, S.; DAVOINE, F.; MASSON, M. A solution for facial expression representation and recognition. *Signal Processing: Image Communication*, v. 17, n. 9, p. 657–673, 2002. ISSN 09235965.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification*. [S.l.]: Wiley, 2001. 654 p. (Pattern Classification and Scene Analysis: Pattern Classification, 6). ISSN 1740634X. ISBN 0471056693.

EKMAN, P. All emotions are basic. *The nature of emotion: Fundamental questions*, 1994.

EKMAN, P.; FRIESEN, W. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, p. 124–129, 1971.

EKMAN, P.; FRIESEN, W. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Palo Alto: Consulting Psychologists Press, 1978.

EKMAN, P.; HEIDER, K. The universality of a contempt expression: A replication. *Motivation and Emotion*, 1988.

FASEL, B.; LUETTIN, J. Automatic facial expression analysis: a survey. *Pattern Recognition*, v. 36, n. 1, p. 259–275, 2003. ISSN 00313203.

FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936. ISSN 00034800.

FREUND, Y.; SCHAPIRE, R.; ABE, N. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, v. 14, n. 5, p. 771–780, 1999.

FREUND, Y.; SCHAPIRE, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, v. 55, n. 1, p. 119–139, 1997. ISSN 00220000.

FUKUNAGA, K. *Introduction to Statistical Pattern Recognition, Second Edition (Computer Science & Scientific Computing)*. 2. ed. [S.l.]: Academic Press, 1990. Hardcover. ISBN 0122698517.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. [S.l.]: Prentice Hall, 2007. 976 p. ISBN 013168728X.

HOFMANN, B. T.; SCH, B.; SMOLA, A. J. KERNEL METHODS IN MACHINE LEARNING 1 Â¨ lkopf By Thomas Hofmann, Bernhard Sch o and Alexander J. Smola. *Annals of Statistics*, v. 36, n. 3, p. 1171–1220, 2008. ISSN 00905364.

HSU, C.-w.; CHANG, C.-c.; LIN, C.-j. A practical guide to support vector classification. *Bioinformatics*, Citeseer, v. 1, n. 1, p. 1–16, 2010.

KANADE, T.; COHN, J. F.; TIAN, Y. T. Y. Comprehensive database for facial expression analysis. *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition Cat No PR00580*, IEEE Comput. Soc, v. 4, p. 46–53, 2000.

KEERTHI, S. S.; LIN, C.-J. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation*, v. 15, n. 7, p. 1667–89, 2003. ISSN 0899-7667.

KIVINEN, J.; WARMUTH, M. K.; AUERC, P. Artificial Intelligence The Perceptron algorithm versus Winnow : linear versus logarithmic mistake bounds when few input variables are relevant. *Artificial Intelligence*, v. 97, n. 97, p. 325–343, 1997.

KOTSIA, I.; BUCIU, I.; PITAS, I. An analysis of facial expression recognition under partial facial image occlusion. *Image Vision Comput.*, Butterworth-Heinemann, Newton, MA, USA, v. 26, n. 7, p. 1052–1067, 2008. ISSN 0262-8856.

KOTSIA, I.; PITAS, I. Facial expression recognition in image sequences using geometric deformation features and Support Vector Machines. *IEEE Transactions on Image Processing*, v. 16, n. 1, p. 172–187, 2007. ISSN 10577149.

LAJEVARDI, S. M.; HUSSAIN, Z. M. Local feature extraction methods for facial expression recognition. *Signal Processing*, Citeseer, v. 3, n. Eusipco, p. 60–64, 2009.

LERNER, B. et al. On the Initialisation of Sammon's Nonlinear Mapping. *Pattern Analysis & Applications*, v. 3, n. 1, p. 61–68, 2000. ISSN 1433-7541.

LIEN, J. J. J. et al. *Subtly different facial expression recognition and expression intensity estimation*. [S.l.]: IEEE, 1998. 853–859 p.

LONGHI, M. T.; BERCHAT, M.; BEHAR, P. A. Reconhecimento de estados afetivos do aluno em ambientes virtuais de aprendizagem. *Revista Novas Tecnologias na Educação*, v. 5, n. 2, 2007.

LUCEY, P. et al. The Extended Cohn-Kanade Dataset (CK +): A complete dataset for action unit and emotion-specified expression. *Image Rochester NY*, IEEE, n. July, p. 94–101, 2010.

LYONS, M. et al. Coding facial expressions with gabor wavelets. In: *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on.* [S.l.: s.n.], 1998. p. 200 –205.

LYONS, M.; BUDYNEK, J.; AKAMATSU, S. Automatic classification of single facial images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 21, n. 12, p. 1357 –1362, dezembro 1999. ISSN 0162-8828.

MARTINS, P.; SAMPAIO, J.; BATISTA, J. Facial Expression Recognition using Active Appearance Models. *VISAPP 2008: Proceedings of the Third International Conference on Computer Vision Theory and Applications*, v. 2, p. 123–129, 2008.

MATSUMOTO, D. Ethnic differences in affect intensity, emotion judgments, display rule attitudes, and self-reported emotional expression in an American sample. *Motivation and emotion*, v. 17, n. 2, p. 107–123, 1993.

MEHRABIAN, A. Communication without words. *Psychological Today*, 1968.

MICHEL, P.; KALIOUBY, R. E. Real Time Facial Expression Recognition in Video using Support Vector Machines. *Computer*, ACM Press, v. 2, p. 258, 2003.

PANTIC, M.; PATRAS, I. Detecting facial actions and their temporal segments in nearly frontal-view face image sequences. In: *Proc. IEEE Int'l Conf. on Systems, Man and Cybernetics*. [S.l.: s.n.], 2005. p. 3358–3363.

PANTIC, M.; PATRAS, I. Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. *IEEE transactions on systems man and cybernetics Part B Cybernetics a publication of the IEEE Systems Man and Cybernetics Society*, v. 36, n. 2, p. 433–449, 2006. ISSN 10834419.

PANTIC, M.; ROTHKRANTZ, L. Automatic analysis of facial expressions: the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 22, n. 12, p. 1424 –1445, dezembro 2000. ISSN 0162-8828.

PANTIC, M.; ROTHKRANTZ, L. J. M. *Facial action recognition for facial expression analysis from static face images.* [S.l.], 2004. v. 34, n. 3, 1449–1461 p.

PASSARINHO C. J. P. ;SALLES, E. O. T. . S. F. M. Face Tracking Framework Using Face Detection in Color Image Multi View with Multi Skin Tones. *XIX Congresso Brasileiro de Automática*, p. 4057–4063, 2012.

PEDROSO, F. J. C.; SALLES, E. O. T. Reconhecimento de expressões faciais baseado em modelagem estatística. *XIX Congresso Brasileiro de Automática*, p. 631–638, setembro 2012.

PERVEEN, N.; GUPTA, S.; VERMA, K. Facial expression recognition using facial characteristic points and gini index. In: *Engineering and Systems (SCES), 2012 Students Conference on.* [S.l.: s.n.], 2012. p. 1 –6.

SAMMON, J. W. A nonlinear mapping for data structure analysis. *Computers, IEEE Transactions on*, C, n. 5, 1969.

SATYANARAYANAN, M. Pervasive computing: vision and challenges. *IEEE Personal Communications*, v. 8, n. 4, p. 10–17, 2001. ISSN 10709916.

SHIH, F. Y.; CHUANG, C.-F.; WANG, P. S. P. Performance comparisons of facial expression recognition in JAFFE database. *Internacional Journal of Pattern Recognition and Artificial Inteligence*, v. 22, n. 3, p. 445–459, 2008.

SHINOHARA, Y.; OTSU, N. Facial expression recognition using fisher weight maps. In: *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*. [S.l.: s.n.], 2004. p. 499 – 504.

SONG, K.-T.; CHEN, Y.-W. A design for integrated face and facial expression recognition. In: *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*. [S.l.: s.n.], 2011. p. 4306 –4311. ISSN 1553-572X.

STEGMANN, M.; GOMEZ, D. A brief introduction to statistical shape analysis. *Informatics and Mathematical Modelling, Technical University of Denmark, DTU*, março 2002.

THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition*. 4th. ed. [S.l.]: Academic Press, 2009. 961 p. ISBN 9781597492720.

THEODORIDIS, S.; KOUTROUMBAS, K. *An introduction to Pattern Recognition: A Matlab Approach*. [S.l.]: Academic Press, 2010. 216 p. ISBN 9780123744869.

TIAN, Y.-I.; KANADE, T.; COHN, J. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, n. 2, p. 97–115, 2001. ISSN 01628828.

VIOLA, P.; JONES, M. Robust real-time object detection. In: *International Journal of Computer Vision*. [S.l.: s.n.], 2001.

WANG, J.; YIN, L. Static topographic modeling for facial expression recognition and analysis. *Comput. Vis. Image Underst.*, Elsevier Science Inc., New York, NY, USA, v. 108, n. 1-2, p. 19–34, 2007. ISSN 1077-3142.

ZHANG, S.; ZHAO, X.; LEI, B. Robust Facial Expression Recognition via Compressive Sensing. *Sensors (Peterboroug)*, v. 12, n. 3, p. 3747–3761, 2012. ISSN 14248220.

ZHANG, Z. et al. Comparison Between Geometry-Based and Gabor-Wavelets-Based Facial Expression Recognition Using Multi-Layer Perceptron. *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, p. 454–459, 1998.

ZHENG, W. Z. W. et al. Facial expression recognition using kernel canonical correlation analysis (KCCA). *IEEE Transactions on Neural Networks*, IEEE, v. 17, n. 1, p. 233–238, 2006.