

**GABRIEL CHAVES AFONSO COUTINHO**

**RECONHECIMENTO FACIAL BASEADO EM SUPPORT VECTOR  
MACHINES**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS  
MINAS GERAIS – BRASIL  
2007

**GABRIEL CHAVES AFONSO COUTINHO**

**RECONHECIMENTO FACIAL BASEADO EM SUPPORT VECTOR  
MACHINES**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:

Inteligência Artificial

Orientador:

Prof. Cristiano Leite de Castro

LAVRAS  
MINAS GERAIS – BRASIL  
2007

**Ficha Catalográfica preparada pela Divisão de Processos Técnico  
da Biblioteca Central da UFLA**

Coutinho, Gabriel Chaves Afonso

Reconhecimento Facial Baseado em Support Vector Machines / Gabriel Chaves Afonso Coutinho. Lavras – Minas Gerais, 2007.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Informática. 2. Reconhecimento Facial. 3. Máquina de Aprendizado. I. COUTINHO G. C. A. II. Universidade Federal de Lavras. III. Título

**GABRIEL CHAVES AFONSO COUTINHO**

**RECONHECIMENTO FACIAL BASEADO EM SUPPORT VECTOR  
MACHINES**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 17 de Janeiro de 2008

---

Prof. André Vital Saúde

---

Prof. Thiago de Souza Rodrigues

---

Prof. Cristiano Leite de Castro  
(orientador)

LAVRAS  
MINAS GERAIS – BRASIL  
2007

## **RESUMO**

Máquina de Vetores de Suporte (SVM) é uma máquina de aprendizagem capaz de resolver problemas de classificação com boa capacidade de generalização. A velocidade de classificação da SVM e a facilidade com que novos vetores de dados podem ser inseridos no conjunto de treinamento dessa máquina virtual são seus pontos mais fortes.

Neste trabalho, é desenvolvido um método para reconhecimento de faces baseado na máquina de vetores de suporte em conjunto com um algoritmo conhecido como Transformada Co-seno Discreta, que compacta as imagens analisadas em vetores de características que podem ser processados pela SVM.

Os experimentos foram realizados com uma base de dados de faces da Universidade de Yale. Os resultados obtidos foram satisfatórios e comprovados pela acurácia do método.

## **ABSTRACT**

Support Vector Machine (SVM) is a learning machine able to solve classification problems. Its classification speed and the easy way new data vectors can be added to the training set are some of the main characteristics of this machine.

In this paper a method for face recognition is developed using Support Vector Machines and a method for compressing images termed Discrete Cosine Transform (DCT). The DCT method was used not only for compressing images but also to transform them into input vectors usable on the SVM.

Experiments were made using the Yale Faces Database. The obtained results were found to be satisfactory, given the accuracy calculated.

# SUMÁRIO

LISTA DE FIGURAS.....	vii
LISTA DE TABELAS.....	viii
1 INTRODUÇÃO.....	1
1.1 Contextualização e Justificativa.....	1
1.2 Motivação.....	2
1.3 Objetivos.....	3
1.4 Organização do Texto.....	3
2 RECONHECIMENTO FACIAL.....	4
2.1 Introdução.....	4
2.2 Problemas Relevantes ao Reconhecimento Facial.....	5
2.2.1 Mudanças na Iluminação.....	5
2.2.2 Mudanças Geométricas.....	6
2.3 A Transformada Co-seno Discreta (DCT).....	7
3 SUPPORT VECTOR MACHINES.....	11
3.1 Aprendizagem de Máquina.....	11
3.2 Teoria do Aprendizado Estatístico.....	12
3.2.1 Risco Funcional.....	13
3.2.2 Minimização do Risco Empírico.....	13
3.2.3 Dimensão VC.....	14
3.2.4 Minimização do Risco Estrutural.....	15
3.3 Support Vector Machines.....	16
3.3.1 Formulação Teórica da SVM.....	16
3.3.2 Problema Primal.....	18
3.3.3 Problema Dual.....	18
3.3.4 Kernel.....	19
3.3.5 Vetores de Suporte.....	19
3.3.6 Exemplo: O problema do XOR.....	20
4 METODOLOGIA.....	25
4.1 A máquina de Vetores de Suporte.....	25
4.2 O Banco de Faces.....	26
4.3 Usando as Faces na SVM-Multiclass.....	26

4.4 Experimentos Realizados.....	27
5 RESULTADOS.....	29
6 CONCLUSÕES E TRABALHOS FUTUROS.....	33
7 REFERÊNCIAS BIBLIOGRÁFICAS.....	34

## LISTA DE FIGURAS

Figura 1.1 – Porcentagem de trabalhos usando diferentes métricas de reconhecimento de pessoas.....	1
Figura 2.1 – Normalização Geométrica.....	6
Figura 2.2 – Face com uma rotação tridimensional.....	7
Figura 2.3 – Uma imagem e sua reconstrução a partir de um vetor DCT.....	10
Figura 3.1 – Dimensão VC.....	14
Figura 3.2 – Superfície ótima de separação entre duas classes.....	16
Figura 3.3 – Vetores de Suporte em um hiper-plano ótimo para padrões linearmente separáveis.....	20
Figura 4.1 – Fotos de uma pessoa do banco de Faces de Yale.....	26



## LISTA DE TABELAS

Tabela 3.1 – Exemplos de Kernel comumente usados.....	19
Tabela 3.2 – Vetores de entrada e as saídas esperadas para o problema do XOR.....	21
Tabela 4.1 – Resultados do pré teste usado para definir o valor de C.....	28
Tabela 5.1 – Resultados do conjunto de aprendizagem e teste 1.....	29
Tabela 5.2 – Resultados do conjunto de aprendizagem e teste 2.....	29
Tabela 5.3 – Resultados do conjunto de aprendizagem e teste 3.....	29
Tabela 5.4 – Resultados do conjunto de aprendizagem e teste 4.....	30
Tabela 5.5 – Resultados do conjunto de aprendizagem e teste 5.....	30
Tabela 5.6 – Resultados do conjunto de aprendizagem e teste 6.....	30
Tabela 5.7 – Resultados do conjunto de aprendizagem e teste 7.....	30
Tabela 5.8 – Resultados do conjunto de aprendizagem e teste 8.....	31
Tabela 5.9 – Resultados do conjunto de aprendizagem e teste 9.....	31
Tabela 5.10 – Resultados do conjunto de aprendizagem e teste 10.....	31
Tabela 5.11 – Resultados do conjunto de aprendizagem e teste 11.....	31
Tabela 5.12 – Resultado geral por valor de C.....	32

# 1 INTRODUÇÃO

## 1.1 Contextualização e Justificativa

Existem diversas tentativas de criar sistemas capazes de identificar pessoas, os mais comuns são baseados nas digitais ou na íris de cada pessoa, mas há ainda sistemas baseados no formato da mão e dos dedos, na voz, assinatura ou na face. Na Figura 1.1 temos um exemplo do uso destas métricas em sistemas de reconhecimento.

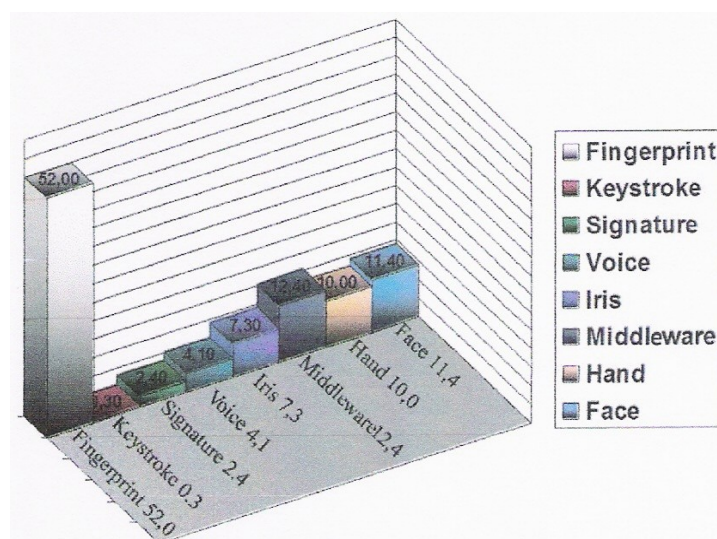


Figura 1.1: Porcentagem de trabalhos utilizando as métricas mais comuns para identificação de pessoas.

Imagem extraída de Abate et al (2007).

Comparando o reconhecimento facial com o reconhecimento de íris ou com as impressões digitais por exemplo, temos a vantagem de que o reconhecimento facial não necessita que um indivíduo colabore ou até mesmo saiba que está sendo identificado. Esta característica, do uso de faces no reconhecimento de pessoas, pode é claro gerar problemas com invasão de privacidade. É portanto necessário ter cuidado onde e quando aplicar tal tecnologia.

Zhao (2003) enumera algumas aplicações onde poderíamos utilizar reconhecimento facial, por exemplo em vídeo games, realidade virtual, interação entre humanos e máquinas, identidade ou passaporte, *logon* em aplicações pessoais, fichas médicas, vídeos de segurança, enfim, aplicações nas mais diversas áreas.

Como será explicado mais adiante, no Capítulo 2, o reconhecimento facial pode ser feito de duas maneiras, através da extração de características de uma foto, ou através de métodos holísticos, que utilizam toda uma foto como fonte de dados. Este trabalho desenvolve um método holístico, utilizando um algoritmo de compressão de dados conhecido como Transformada Co-seno Discreta (DCT) para transformar uma imagem em um vetor de características. Tais valores são utilizados como entrada em uma máquina de aprendizagem conhecida como Máquina de Vetores de Suporte (SVM) que tem apresentado bons resultados em tarefas de reconhecimento de padrão

## 1.2 Motivação

A acurácia das Support Vector Machines (SVMs) na tarefa de reconhecimento de padrões é comprovada por inúmeros trabalhos na literatura (VAPNIK 1999).

As SVMs são baseadas no princípio de minimização do risco estrutural através de um algoritmo de aprendizagem que realiza a maximização da margem entre as classes de dados do problema. Dessa forma, as SVMs obtêm boa capacidade de generalização.

Na tarefa de reconhecimento de imagens faciais o principal problema é a alta dimensionalidade dos dados de entrada (imagens). Dessa forma, é necessário que seja aplicado um esquema de codificação e redução de dimensionalidade das imagens para que essas possam ser utilizadas como vetores de entrada para as SVMs. É importante que esse esquema de redução de dimensionalidade preserve as principais características da imagem de modo que a tarefa de reconhecimento facial não seja prejudicada.

Nos trabalhos de Hafed e Levine (2001) e Er et al (2005) têm-se observado que o método de compressão de dados Transformada Co-seno Discreta (DCT) tem obtido bons resultados na extração de características de imagens faciais. Observa-se que com a aplicação desse método as principais características se conservam nos coeficientes de menor frequência da imagem.

Baseado nessas presentes características, o presente trabalho propõe um método de reconhecimento facial que incorpora o método DCT para a redução de dimensionalidade das faces e sua transformação em vetores de características que podem ser utilizadas como vetores de entrada para as SVMs cuja tarefa é identificar as faces dos indivíduos no conjunto de trabalho.

## 1.3 Objetivos

O objetivo geral deste trabalho é desenvolver e testar um método de reconhecimento facial utilizando o método DCT e a máquina de aprendizado SVM.

Destacam-se como objetivos específicos:

- 1 – Fazer uma revisão de literatura sobre reconhecimento facial
- 2 – Fazer uma revisão de literatura sobre SVMs e DCT
- 3 – Implementar um método de reconhecimento facial baseado em DCT e SVM
- 4 – Aplicar o método desenvolvido em uma base de dados de faces disponível na literatura
- 5 – Analisar os resultados obtidos.

## 1.4 Organização do Texto

No Capítulo 2 é apresentado o estado da arte do Reconhecimento Facial, alguns problemas comuns ao se trabalhar com imagens faciais e também o método de Transformada Co-seno Discreta (DCT).

No Capítulo 3 são discutidos, o conceito de Aprendizagem de Máquinas, e a teoria do aprendizado estatístico, ambos importantes para a compreensão da formulação teórica da máquina de vetores de suporte. O funcionamento da SVM é explicado, e um exemplo de um problema de classificação não linearmente separável é ilustrado.

O capítulo 4, traz a metodologia usada para a implementação do método de reconhecimento facial empregado. Além disso, é feita uma descrição da base de faces utilizada nos experimentos realizados.

No capítulo 5 são apresentados os resultados obtidos na forma de tabelas, e também a acuidade do método de reconhecimento proposto neste trabalho para o banco de faces da universidade de Yale.

No capítulo 6 é feita uma breve análise sobre os dados obtidos no capítulo 5, e trabalhos futuros são sugeridos com base nestes dados.

# 2 RECONHECIMENTO FACIAL

## 2.1 Introdução

Existem dois grandes motivos para pesquisas na área de reconhecimento facial estarem se tornando mais comuns; o primeiro é o grande número de aplicações que podem ser desenvolvidas, principalmente em se tratando de aplicações que lidam com segurança, e o segundo são as tecnologias que têm se tornado cada vez mais aptas para lidar com problemas de alto custo computacional que antes era considerado proibitivo.

Estes estudos são empregados principalmente por estudiosos em áreas como processamento de imagens, reconhecimento de padrões, redes neurais artificiais, visão computacional e psicologia.

Zhao et al (2003) define um problema de reconhecimento facial como: Dada uma cena, seja ela vídeo ou foto, localizar e identificar uma ou mais pessoas em cena baseado em um conjunto de faces. Para isto podem ser usadas informações auxiliares, como raça, idade, sexo, expressão facial, ou outras. Ele ainda divide o problema em três etapas:

- Detecção de faces: consiste em localizar em uma imagem o rosto das pessoas presentes na cena.

- Extração de Características: esta etapa trata de, dada uma face, recolher os dados que serão usados pela rede de reconhecimento facial. Pode envolver também o reconhecimento de emoções faciais e estimativas a respeito da direção dos olhos da face.

- Por último, temos o Reconhecimento Facial em si, que compara as características extraídas da cena com as características das faces armazenadas em banco, e define se a face encontrada na cena pertence a alguma pessoa conhecida pelo sistema, e em caso afirmativo aponta que pessoa é essa.

Como o problema é multi disciplinar, classificar uma solução de acordo com as áreas em que ela se apóia para classificação e extração de características é uma tarefa difícil (ZHAO 2003). Para ter uma divisão mais precisa sobre as soluções desenvolvidas, elas são classificadas de acordo com estudos psicológicos que tentam desvendar se os humanos usam um sistema de reconhecimento baseado em extração de características de uma face ou usam as faces como um todo (método holístico):

- Métodos Holísticos: Estes métodos usam toda a face como matéria prima para o sistema de reconhecimento. O método holístico mais conhecido é o método das *Eigenpictures* (Kirby e Sirovich 1990).

- Métodos baseados na Extração de Características: Estes métodos são normalmente baseados em pontos chave da face, como olhos, nariz e boca e sua posição ou aparência, e estes dados são passados para a máquina de classificação.

- Métodos Híbridos: Assim como o sistema humano usa ambos os métodos holísticos e de extração de características. Uma máquina também pode combiná-los para utilizar o ponto forte de cada um.

Métodos baseados em extração de características foram os mais comuns nas primeiras tentativas de automatizar o processo, utilizando idéias mais simples, como detecção de fronteiras para localizar as características da imagem (SAKAI 1969 apud HAFED 2001). Um estudo feito por Brunelli e Poggio (1993) sobre métodos baseados em extração de características faz uma comparação entre esses métodos e métodos holísticos.

## **2.2 Problemas Relevantes ao Reconhecimento Facial**

Como o reconhecimento facial é feito com base em uma foto, uma série de problemas surge dificultando o trabalho das máquinas de aprendizado e algoritmos usados neste reconhecimento. Existem, entretanto, algumas maneiras de amenizar esses problemas

### **2.2.1 Mudanças na Iluminação**

Mudanças na Iluminação são capazes de confundir os algoritmos de reconhecimento a ponto de considerarem uma imagem com iluminação invertida por exemplo como uma outra pessoa (ZHAO et al 2003). Para corrigir diferenças na iluminação são desenvolvidos algoritmos de normalização da iluminação. Estes algoritmos são normalmente baseados em criar uma paleta de cores padrão e realizar transformações sobre uma foto para que ela adote as cores desta paleta. Este algoritmo é conhecido como histograma de Hummel (HAFED et al 2001).

Um estudo de Hafed (1996) apud Hafed (2001) mostrou que a melhor maneira de

tratar a normalização da iluminação é compensar as diferenças de iluminação entre as duas metades da face em questão, deste modo cada foto tem sua própria paleta de cores, evitando a perda de informação quanto a cor de pele da face por exemplo.

## 2.2.2 Mudanças Geométricas

Normalizações geométricas tem por objetivo transformar as faces obtidas de modo que todas possuam a mesma escala e a mesma rotação. Além disso, estas normalizações podem também excluir o restante da imagem que não é parte da face.

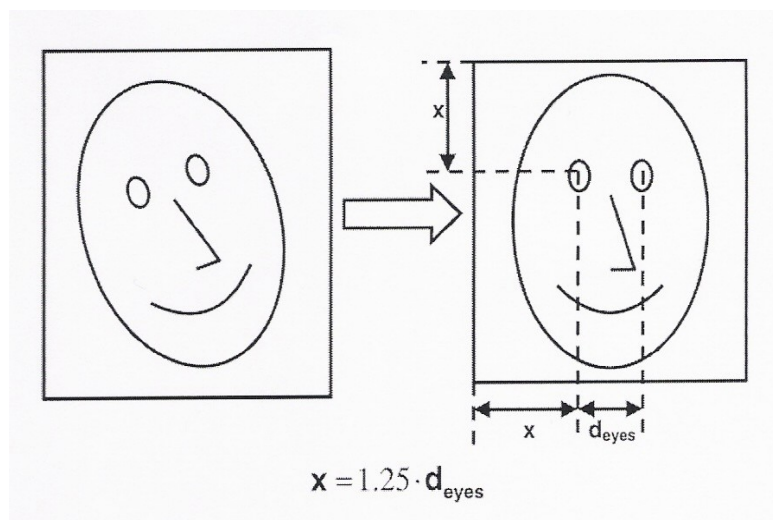


Figura 2.1: Normalização geométrica e os parâmetros usados. Imagem extrapida de Hafed e Levine (2001).

Vale ressaltar que neste método apenas as alterações bidimensionais na imagem são corrigidas. Hafed (2001) sugere o uso de fotos de um mesmo indivíduo em diversas posições como uma maneira de compensar pelo não tratamento de alterações tridimensionais na posição da face na imagem. A Figura 2.2 mostra um exemplo de uma face com alterações tridimensionais.

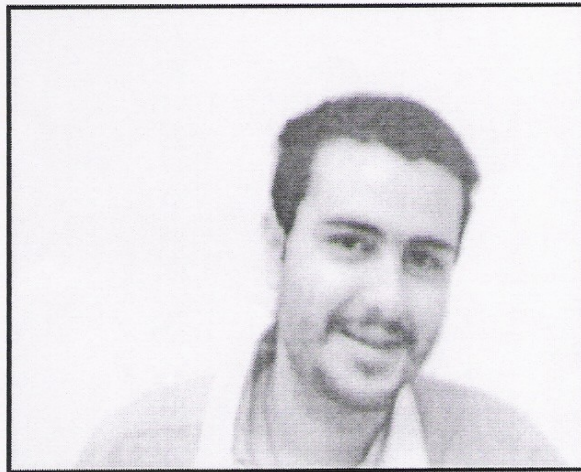


Figura 2.2: Face com uma rotação tridimensional que não pode ser corrigida com as transformações sugeridas na Figura 2.1 para gerar uma imagem correta. Imagem extraída de Hafed e Levine (2001).

Abate et al (2007) cita métodos que consideram alterações tridimensionais no reconhecimento facial e destaca o método descrito em Gross et al (2002) apud Abate (2007), que utiliza *eigenfaces* geradas a partir de imagens de diversos indivíduos em posições diferentes.

## 2.3 A Transformada Co-seno Discreta (DCT)

A transformada co-seno discreta é um método de compressão de dados utilizado na compressão de imagens JPG (Join Photographic Experts Group).

A compressão de dados é importante tanto para o processo de reconhecimento biológico quanto para o computacional. Segundo Sekuler e Blake (1994) apud Hafed e Levine (2001), na retina humana há uma compressão de 100:1 dos sinais recebidos.

Segundo Hafed e Levine (2001), o DCT é classificado em quatro diferentes transformações, DCT-I, DCT-II, DCT-III e DCT-IV. Neste trabalho utilizamos a segunda transformação, DCT-II.

Considerando uma imagem como um vetor  $u(n)$  de tamanho  $N$ , podemos obter o vetor DCT  $v(k)$  através da seguinte equação:

$$v(k) = \alpha(\kappa) \sum_{n=0}^{N-1} u(n) \cos\left(\frac{(2n+1)\pi\kappa}{2N}\right)$$
$$0 \leq k \leq N-1$$



(2.1a)

onde

$$\alpha(0) = \sqrt{\frac{1}{N}}, \quad \alpha(\kappa) = \sqrt{\frac{2}{N}} \quad 1 \leq \kappa \leq N-1$$

(2.1b)

Uma outra maneira de entender o método DCT é considerar a seqüência  $u(n)$  como um vetor, e o método DCT como uma transformação matricial que aplicada a este vetor gera o vetor  $v(k)$ . Neste caso a transformação matricial DCT,  $C = \{c(k,n)\}$  é definida da seguinte forma:

$$c(k,n) = \left\{ \begin{array}{l} \frac{1}{\sqrt{N}} \quad k=0, 0 \leq n \leq N-1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2n+1)\pi\kappa}{2N}\right) \quad 1 \leq k \leq N-1 \text{ e } 0 \leq n \leq N-1 \end{array} \right\}$$

(2.2)

onde  $k$  e  $n$  são os índices de linha e coluna respectivamente. Usando a equação 2.2, o DCT da seqüência  $u(n)$  será:

$$v = Cu$$

(2.3)

Para obtermos a imagem novamente temos que realizar a operação inversa da transforma co-seno discreta:

$$u(n) = \sum_{\kappa=0}^{N-1} \alpha(\kappa) v(\kappa) \cos\left(\frac{(2n+1)\pi\kappa}{2N}\right)$$

$$0 \leq n \leq N-1$$

(2.4)

onde  $\alpha(k)$  é dado na equação (2.1b).

Se usarmos a equação 2.3, podemos obter a sequencia  $u(n)$  a partir de um vetor DCT  $v(n)$  usando a equação:

$$u = C^{-1}v$$

(2.5)

Na compressão de imagens JPG (Join Photographic Experts Group), a imagem é dividida em sub-blocos de 8x8 pixels e o algoritmo é aplicado em cada um desses blocos, e sempre é gerada uma matriz DCT com o mesmo tamanho da imagem, ou seja, em uma imagem de dimensões MxN, é gerada uma matriz DCT MxN (ER et al 2005).

Neste trabalho o método DCT é aplicado sobre toda a imagem e não em sub-blocos porque estamos usando uma abordagem holística, que considera a imagem como um todo, e para cada imagem é gerada uma matriz 8x8. Através do método DCT as características mais significativas da imagem são extraídas. O tamanho desta matriz é sugerido por Hafed e Levine(2001) por trabalhar apenas com características de baixa e média frequência da imagem, o que é sugerido também em Er (2005):

*Como um método de extração de características holístico, o DCT converte imagens faciais de altas dimensões em espaços de dimensões menores, onde características faciais mais importantes como contorno dos cabelos e da face, posição dos olhos nariz e boca são mantidos. Essas características faciais são mais estáveis do que as características faciais de alta frequência. De fato, o sistema visual humano é mais sensível a alterações na banda de baixa frequência.*

Como resultado a transformação de uma imagem em um vetor DCT 8x8 há grande perda de dados, e não é mais possível reconstruir a imagem original, como pode ser visto na figura 2.3. Isto entretanto, não é um problema para o método de reconhecimento. Nos trabalhos de Turk e Pentlans (1991) utilizando um outro método de extração de características conhecido como KLT (Karhunen-Loève Transform) que trabalha com *eigenfaces*, muitas vezes os coeficientes usados não eram suficientes para fazer a reconstrução da imagem, mas os resultados obtidos no reconhecimento das faces foram bons. Aumentar o tamanho do vetor DCT não implica em melhores resultados no processo de reconhecimento facial.



Figura 2.3: Uma imagem e sua reconstrução a partir de um vetor DCT de 64 posições (8x8). Imagem extraída de Hafed e Levine (2001).

# 3 SUPPORT VECTOR MACHINES

## 3.1 Aprendizagem de Máquina

Máquinas que podem aprender se tornaram importantes com a grande quantidade de problemas reais para os quais não existem soluções algorítmicas exatas, ou para os quais estas soluções não são factíveis. Enquanto o desenvolvimento de um programa convencional requer um conhecimento detalhado do domínio destes problemas, uma máquina de aprendizado requer apenas que exemplos sejam apresentados para que ela possa solucioná-los (Mjolsness & DeCoste 2001 apud Carvalho 2005).

Mendel e McClaren (1970) apud Haykin (1999) definem aprendizagem no contexto de redes neurais artificiais (RNAs) como:

*“Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre.”*

Vapnik (1995) apud Carvalho (2005) define que uma máquina de aprendizado, após a observação de vários pares de entrada  $X$  e saída  $Y$  deve ser capaz de:

- Imitar o comportamento do sistema, gerando, a partir de dados de entrada próximos a  $X$ , saídas próximas a  $Y$ . Esta tarefa é conhecida como classificação, pois o domínio de  $X$  é dividido de acordo com o número de saídas  $Y$  possíveis no sistema.

- Descobrir a função que governa o sistema. Neste caso o número de saídas  $Y$  é infinito, e esta tarefa é conhecida como regressão.

Para que uma máquina seja capaz de realizar uma das tarefas acima, ela deve ser submetida a um processo de aprendizagem, também conhecido como processo de treinamento.

Existem diversas maneiras de desenvolver um programa que torne o computador capaz de aprender, as Support Vector Machines (SVMs) se baseiam na teoria do aprendizado estatístico.

## 3.2 Teoria do Aprendizado Estatístico

As Support Vector Machines são baseadas num tipo de aprendizado chamado de aprendizado supervisionado, ou aprendizado com professor. Este aprendizado supervisionado consiste de três componentes (Vapnik, 1992, 1998 apud Haykin, 1999):

1. Ambiente: conjunto de vetores de entrada  $x$ .
2. Professor: O professor fornece uma resposta  $d$  para cada vetor de entrada  $x$  recebido de acordo com uma função  $f(x)$  desconhecida.
3. Máquina: Ou algoritmo de aprendizagem, é capaz de implementar funções de mapeamento de entrada-saída da forma  $y = f(x, r)$  onde  $y$  é a resposta produzida pela máquina e  $r$  é um conjunto de parâmetros usados como pesos aos valores do vetor  $x$ .

Os dados do conjunto de treinamento devem ser estatisticamente representativos, para que a máquina possa reconhecer possíveis padrões posteriores não apresentados a ela inicialmente. Esta é uma propriedade conhecida como Generalização (Vapnik 1982 apud Carvalho, 2005).

Além desta grande amostra de dados, é necessário também que as funções  $d = F(x, r)$  tenham comportamento determinístico, ou seja, para um certo conjunto de entrada  $X$ , e um conjunto de parâmetros  $r$ , a saída deve ser sempre a mesma.

O objetivo da máquina de aprendizado é escolher uma função  $f(x, r)$  que seja capaz de mapear a relação de  $x$  e  $y$ , onde  $r$  são os parâmetros desta relação. As funções usadas para aprender este mapeamento são conhecidas como funções indicadoras em problemas de classificação e de funções de aproximação em problemas de regressão (Vapnik 1999).

Para escolher a melhor função que se ajuste ao conjunto de treinamento é necessária uma medida de discrepância  $L(y, f(x, r))$ , que indica a diferença da saída desejada  $d$  da saída obtida  $y$  (Vapnik 1982 apud Carvalho 2005).

Para problemas de classificação binária, com somente duas classes, são usadas funções de discrepância como:

$$L(y, f(x, r)) = \begin{cases} 0, & \text{se } y = f(x, r) \\ 1, & \text{se } y \neq f(x, r) \end{cases} \quad (3.1)$$

### 3.2.1 Risco Funcional

O valor esperado para o erro da máquina de aprendizagem, utilizando-se uma função indicadora escolhida, em relação às novas amostras pertencentes à distribuição conjunta de probabilidade  $p(x, y)$ , é chamado risco funcional.

A tarefa da máquina de aprendizagem é encontrar o menor risco funcional sobre o conjunto de amostras  $D$ . Este risco funcional pode ser expresso por:

$$R_{\text{funcional}} = \int L(y, f(x, r)) dP(x, y) \quad (3.2)$$

Conhecendo o valor de  $P(x, y)$  e com uma função de perda devidamente escolhida, a integral pode ser resolvida usando a relação:

$$dP(x, y) = p(x, y) dx dy \quad (3.3)$$

Porém o valor de  $P(x, y)$  não costuma ser conhecido.

### 3.2.2 Minimização do Risco Empírico

O risco empírico é relacionado ao número de amostras usadas no conjunto de aprendizagem, e é representado por:

$$R_{\text{empirico}} = \frac{1}{N} \sum_{i=1}^N L(y, f(x, r)) \quad (3.4)$$

Minimizar o risco empírico costuma acarretar problemas conhecidos como *overfitting* e *underfitting*. *Overfitting* é quando a complexidade de uma função indicadora é superior a complexidade do problema resolvido, e *underfitting* é quando a complexidade da função é inferior a complexidade do problema. Em ambos os casos a capacidade de

generalização da máquina é reduzida (Haykin, 1999).

Segundo Vapnik (1999), quando o número  $N$  de amostras tende a infinito o risco empírico converge para o risco funcional. Na maioria dos casos entretanto, o número de amostras disponíveis não é suficiente para minimizar o risco empírico a ponto de minimizar o risco funcional.

### 3.2.3 Dimensão VC

Denominada de dimensão Vapnik-Chervonenkis ou simplesmente dimensão VC em homenagem a seus criadores Vapnik e Chervonenkis (1971) esta dimensão é uma medida da capacidade de classificação de um grupo de funções indicadoras realizadas pela máquina de aprendizagem.

O valor da dimensão VC equivale ao maior número de exemplos de treinamento que podem ser aprendidos pela máquina sem erros, e é calculado da seguinte forma;  $VC = 2$  quando o problema pode ser separado por uma reta,  $VC = 3$  quando o problema pode ser separado por um plano,  $VC = 4$ , quando o problema é separado por um hiperplano e assim por diante. A dimensão VC é  $n+1$  sendo  $n$  a dimensão do espaço vetorial em questão. Um exemplo do cálculo da dimensão VC é ilustrado na figura 3.1.

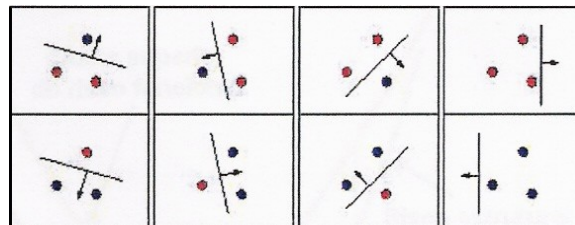


Figura 3.1: Dimensão VC – É quantificada pelo maior número de vetores  $h$  que podem ser separados em duas classes diferentes, para uma dimensão especificada, neste caso  $n = 2$ . Imagem extraída de Carvalho (2005)

Para ajustar adequadamente a complexidade da máquina de aprendizagem em relação ao problema é necessário um princípio indutivo conhecido como minimização do risco estrutural que utiliza a dimensão VC como parâmetro de controle (Vapnik & Lerner 1963 apud Carvalho 2005; Vapnik & Chervonenkis 1964 apud Carvalho 2005). Sua implementação pode ser efetuada por meio da maximização da margem de separação entre as classes.

### 3.2.4 Minimização do Risco Estrutural

Com o uso da dimensão VC foi desenvolvida uma expressão que indica o limite superior do risco funcional, com probabilidade  $1 - e$  de ocorrer, dado um valor  $e$  entre 0 e 1 (Vapnik 1998 apud Carvalho 2005):

$$R_{funcional} \leq R_{empirico} + R_{estrutural}(h, N, e) \quad (3.5)$$

Com esta expressão uma máquina de aprendizagem passa a ser capaz de reduzir o limite superior do risco funcional, o que é o objetivo do processo de aprendizagem.

O termo  $R_{estrutural}(h, N, e)$  pode ser expresso como:

$$R_{estrutural}(h, N, e) = \sqrt{\frac{h \left( \log \left( \frac{2N}{h} \right) + 1 \right) - \log \left( \frac{e}{4} \right)}{N}} \quad (3.6)$$

Onde  $h$  é a dimensão VC,  $N$  o número de amostras de treinamento e  $e$  um parâmetro escolhido.

O objetivo da máquina de aprendizagem é encontrar um valor que alie o risco empírico e o risco estrutural, já que a diminuição do primeiro implica o crescimento do segundo. Com isso obtém-se o menor limite superior possível para o risco funcional.

Um problema da minimização do risco estrutural é que quase nunca se consegue valores exatos para a dimensão VC, esta então costuma ser estimada por meio de limites superiores (Vapnik, Levin & Lecun 1994 apud Carvalho 2005):

$$h \leq 1 + \min \left( N, \left( \frac{R^2}{T^2} \right) \right) \quad (3.7)$$

Onde  $R$  é o menor raio possível de uma hiper-esfera circunscrita aos vetores de



treinamento, e  $T$  é a margem de separação entre as classes. Uma outra forma de minimizar o risco estrutural sugerida por Vapnik (1982) apud Carvalho (2005) é a maximização da margem de separação  $T$ , visto que  $T$  é inversamente proporcional à dimensão VC.

### 3.3 Support Vector Machines

Support Vector Machine (SVM) é uma máquina de aprendizagem baseada na minimização do risco estrutural. Seu processo de aprendizado é do tipo supervisionado e ela pode ser usada na resolução de problemas de classificação e de regressão. Num contexto de classificação binária por exemplo a idéia principal da SVM é construir um hiperplano como superfície de separação ótima entre exemplos positivos e exemplos negativos (Haykin 1999).

O hiper-plano de separação encontrado por uma SVM, e mostrado na figura 3.2 é ótimo pois só existe um ponto que minimiza a função de custo quadrática existente no problema de otimização característico da SVM.

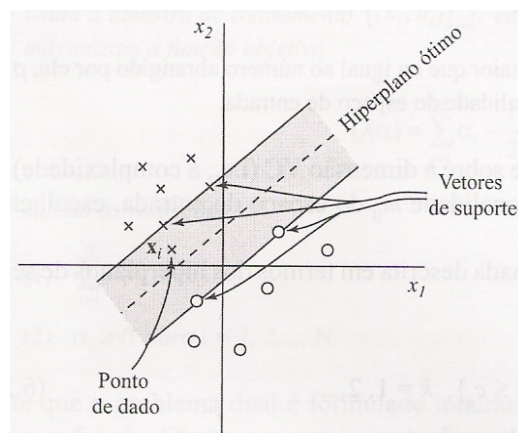


Figura 3.2: Ilustração de uma superfície ótima de separação entre duas classes. O ponto  $x_i$  se encontra dentro da região de separação, mas do lado correto da superfície de decisão. Imagem extraída de Haykin (1999).

#### 3.3.1 Formulação Teórica da SVM

Sendo um conjunto de treinamento  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  onde  $y$  tem valor 1 ou -1, indicando a que classe  $x$  pertence, e  $x$  é um vetor de  $p$  dimensões normalizado (com valores 0, 1 ou -1) a SVM divide o hiper-plano de acordo com a seguinte equação:

$$w^T x + b = 0 \tag{3.8}$$

Onde o vetor  $w$  é um vetor perpendicular ao hiper-plano e  $b$  é uma variável que permite que a margem do hiper-plano seja maximizada, pois sem esta variável o hiper-plano obrigatoriamente passaria pela origem.

Como estamos interessados na maior margem de separação, temos que dar atenção aos hiper-planos paralelos ao hiper-plano ótimo mais próximos aos vetores de suporte de cada classe. Estes hiper-planos podem ser descritos pelas equações:

$$w^T x - b = 1 \tag{3.9}$$

$$w^T x - b = -1 \tag{3.10}$$

Sendo o conjunto de dados de aprendizado linearmente separável, podemos selecionar estes hiper-planos maximizando a distância entre eles de modo que não hajam pontos no intervalo destes pontos. A geometria mostra que a distância entre estes dois hiper-planos é  $2/|w|$ , portanto queremos minimizar o valor de  $|w|$ , garantindo que para cada vetor de entrada  $x_i$ :

$$\begin{cases} w^T x_i + b \geq 1 \text{ se } y_i = 1 \\ w^T x_i + b \leq -1 \text{ se } y_i = -1 \end{cases} \tag{3.11}$$

O que pode ser reescrito como:

$$y_i [w^T x_i + b] \geq 1 \tag{3.12}$$

### 3.3.2 Problema Primal

O problema agora é um problema de minimização, que tem como restrição a equação 3.12:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^T w \\ \text{sujeito a} \quad & y_i [w^T x_i + b] \geq 1 \end{aligned} \tag{3.13}$$

Onde o fator  $\frac{1}{2}$  é usado apenas para facilitar os cálculos.

### 3.3.3 Problema Dual

O problema primal pode ser difícil de resolver devido a natureza das restrições, por isso usa-se a teoria dos lagrangeanos (Bertsekas 1982 apud Carvalho 2005) para criar um modelo conhecido como modelo dual.

O problema Lagrangeano dual pode ser obtido pelo acréscimo das restrições primais à função de custo primal, utilizando multiplicadores chamados de multiplicadores de Lagrange. Os multiplicadores associados às restrições de desigualdade devem ser positivos, enquanto os associados à restrição de igualdade podem assumir quaisquer valores.

O problema dual é descrito da seguinte forma (Haykin 1999):

$$\begin{aligned} \max_{\alpha,\beta} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{sujeito a} \quad & \left\{ \begin{array}{l} \sum_{i=1}^N \alpha_i y_i = 0 \\ 0 \geq \alpha_i \geq C, i = 1, \dots, N \end{array} \right\} \end{aligned} \tag{3.14}$$

Onde  $\alpha_i$  é o multiplicador de Lagrange, e  $C$  um valor de treinamento que limita o valor dos multiplicadores de Lagrange.

### 3.3.4 Kernel

No problema dual existe um produto interno entre vetores  $x_i$  e  $x_j$ . Podemos trocar estes vetores por funções  $K(x_i)$  e  $K(x_j)$ , com isso conseguimos resolver problemas não linearmente separáveis, onde é necessário aumentarmos a dimensão dos vetores de entrada.

$$\begin{aligned} \max_{\alpha, \beta} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i) K(x_j) \\ \text{sujeito a} \quad & \left\{ \begin{array}{l} \sum_{i=1}^N \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, i = 1, \dots, N \end{array} \right\} \end{aligned} \tag{3.15}$$

O produto  $K(x_i).K(x_j)$  pode ser substituído por uma única função conhecida como função de Kernel, e que deve obedecer ao Teorema de Mercer (Mercer 1909 apud Carvalho 2005):

$$K(x_i, x_j) = K(x_j, x_i) = K(x_i).K(x_j) \tag{3.16}$$

Algumas funções de Kernel comumente usadas estão na tabela 3.1:

Tabela 3.1: Exemplos de Kernel comumente usados

Kernel	Equação
Polinomial	$(x_i \cdot x_j + 1)^n$
Rede de função de Base Radial	$e^{-\frac{\ x_i - x_j\ ^2}{2r^2}}$
Sigmóide	$\tanh(\beta_0 x_i^T x_j + \beta_1)$

### 3.3.5 Vetores de Suporte

Na fórmula do modelo dual, os vetores que possuem multiplicadores de Lagrange

não nulos são chamados de vetores de suporte, e são estes vetores que contribuem para a construção do hiper-plano ótimo. Graficamente, os vetores de suporte são os vetores situados dentro do hiper-plano de separação, nos limites do hiper-plano ou no lado errado da superfície e decisão, conforme ilustrado na figura 3.3.

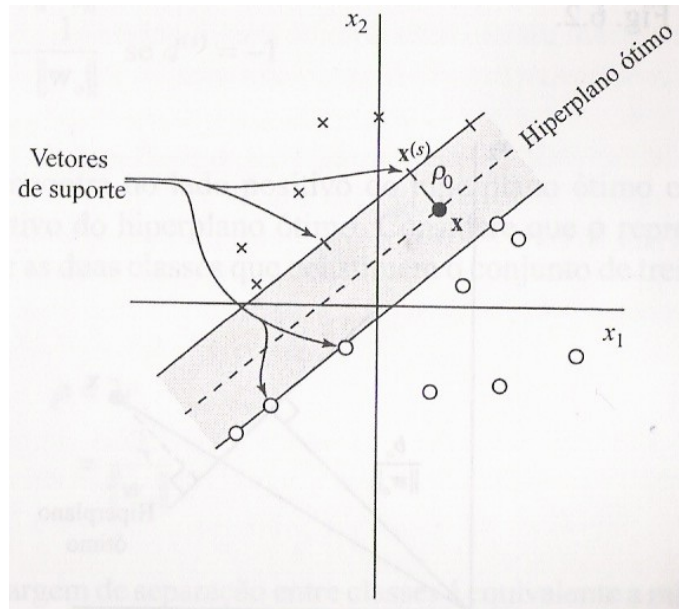


Figura 3.3: Vetores de suporte em um hiper-plano ótimo para padrões linearmente separáveis. Imagem extraída de Haykin (1999).

O valor do multiplicador de Lagrange associado ao vetor no final do processo de treinamento é o que o classifica um vetor de suporte:

- $a = 0$  e  $y_i f(x_i) > 1$  caracterizam um vetor comum, situado na região correta de sua classe.
- $0 < a < C$  e  $y_i f(x_i) > 1$  caracterizam um vetor de suporte, situado sobre a margem mais próxima de sua classe.
- $a_i = C$  e  $y f(x) < 1$  caracterizam um vetor de suporte fora da região de sua classe.

### 3.3.6 Exemplo: O Problema do XOR

Para ilustrar o funcionamento de uma máquina de vetores de suporte foi retirado este exemplo de Haykin (1999) da resolução do problema do XOR utilizando um kernel

polinomial. A tabela 3.2 apresenta um resumo dos vetores de entrada e as respostas desejadas para os quatro estados possíveis.

Tabela 3.2: Vetores de entrada e suas respostas desejadas para o problema do XOR

Vetor de entrada $X$	Resposta desejada $d$
(-1,-1)	-1
(-1,+1)	+1
(+1,-1)	+1
(+1,+1)	-1

Vamos definir o kernel que será usado como:

$$K(x, x_i) = (1 + x^T x_i)^2 \quad (3.17)$$

Com  $x = [x_1, x_2]^T$  e  $x_i = [x_{i1}, x_{i2}]^T$ , podemos assim expressar o núcleo do produto interno  $K(x, x_i)$  em termos de monômios de várias ordens, como segue:

$$K(x, x_i) = 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_2^2 x_{i2}^2 + 2x_1 x_{i1} + 2x_2 x_{i2}$$

A imagem do vetor de entrada  $x$  induzida no espaço de características é:

$$Q(x) = [1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]^T$$

Similarmente:

$$Q(x_i) = [1, x_{i1}^2, \sqrt{2}x_{i1} x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}]^T, \quad i=1,2,3,4$$

Se considerarmos  $k(x_i, x_j)$  como o elemento de uma matriz de  $i$  linha e  $j$  colunas, podemos construir essa matriz  $K$  como:

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

A função objetivo para a forma dual é portanto (segundo a fórmula 3.14):

$$q(a) = a_1 + a_2 + a_3 + a_4 - \frac{1}{2}(9a_1^2 - 2a_1a_2 - 2a_1a_3 + 2a_1a_4 + 9a_2^2 + 2a_2a_3 - 2a_2a_4 + 9a_3^2 - 2a_3a_4 + 9a_4^2)$$

A otimização de  $q(a)$  em relação aos multiplicadores de Lagrange produz o seguinte conjunto de equações simultâneas:

$$\begin{aligned} 9a_1 - a_2 - a_3 + a_4 &= 1 \\ -a_1 + 9a_2 + a_3 - a_4 &= 1 \\ -a_1 + a_2 + 9a_3 - a_4 &= 1 \\ a_1 - a_2 - a_3 + 9a_4 &= 1 \end{aligned}$$

Assim, os valores ótimos dos multiplicadores de Lagrange são:

$$a_{o,1} = a_{o,2} = a_{o,3} = a_{o,4} = \frac{1}{8}$$

Este resultado indica que neste exemplo todos os quatro vetores de entrada são vetores de suporte. O valor ótimo de  $q(a)$  é:

$$q_0(a) = \frac{1}{4}$$

Correspondentemente podemos escrever:

$$\frac{1}{2} \|w_0\|^2 = \frac{1}{4}$$

ou

$$\|w_0\| = \frac{1}{\sqrt{2}}$$

O vetor de pesos ótimo é definido como

$$w_0 = \sum_{i=1}^N a_{0,i} d_i Q(x_i) \tag{3.18}$$

portanto

$$w_0 = \frac{1}{8} [-Q(x_1) + Q(x_2) + Q(x_3) - Q(x_4)]$$

$$w_0 = \frac{1}{8} \left[ \begin{array}{c} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \end{array} \right] + \begin{array}{c} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \end{array} + \begin{array}{c} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \end{array} - \begin{array}{c} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ \sqrt{2} \end{array} = \begin{array}{c} 0 \\ 0 \\ \frac{-1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{array}$$

O hiper-plano ótimo é definido por

$$w_0^T Q(x) = 0 \tag{3.19}$$

Isto é



$$\begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0$$

que se reduz a

$$-x_1x_2 = 0$$

Para  $X_1 = X_2 = -1$  e  $X_1 = X_2 = +1$ , a saída  $-1$ ; e para  $X_1 = -1, X_2 = +1$  e  $X_1 = +1, X_2 = -1$ , temos a saída  $+1$ . Assim o problema do XOR é resolvido.

## 4 METODOLOGIA

### 4.1 A Máquina de Vetores de Suporte

Neste trabalho foi usada uma implementação de uma Support Vector Machine chamada SVM-Multiclass, desenvolvida por Thorsten Joachims, professor do departamento de Ciência da Computação da Universidade de Cornell, e disponível no site [http://www.cs.cornell.edu/People/tj/svm%5Flight/svm\\_multiclass.html](http://www.cs.cornell.edu/People/tj/svm%5Flight/svm_multiclass.html). Esta implementação foi escolhida por que trabalha com problemas de classificação com mais de duas classes.

A SVM-Multiclass utiliza como entrada em ambos os módulos, de treinamento e de teste, um arquivo de texto onde cada linha corresponde a um vetor de dados, ou seja, uma imagem de uma pessoa, e esta linha deve estar no seguinte formato:

```
<classe> <característica>:<valor> <característica>:<valor>...
```

onde a classe é um número usado para identificar cada pessoa; a característica corresponde a cada coeficiente do vetor DCT gerado a partir da foto do indivíduo. Cada imagem foi transformada em um vetor com 64 características (coeficientes) DCT. Cada indivíduo corresponde a uma classe.

A execução dos módulos da SVM-Multiclass é feita através de linhas de comando, para o módulo de treinamento esta linha é escrita da seguinte forma:

```
svm_multiclass_learn -c <C> <arquivo de entrada> <arquivo de saída>
```

onde C representa o parâmetro do limite dos multiplicadores de lagrange nas equações 3.14 e 3.15; arquivo de entrada é o arquivo de vetores para treinamento, e arquivo de saída é um arquivo que será usado pelo módulo de teste e que guarda os parâmetros usados pela SVM após o treinamento.

A linha de comando para a execução do módulo de classificação é a seguinte:

`svm_multiclass_classify <arq. de entrada 1> <arq. de entrada 2> <arq. de saída>`

onde arquivo de entrada 1 é o arquivo com os vetores que serão classificados, arquivo de entrada 2 é o arquivo de saída do módulo de treinamento e arquivo de saída é um arquivo onde cada linha representa um vetor do arquivo de classificação e contém o número da classe a qual a SVM considera que o vetor pertence, ou seja, o resultado da classificação para o conjunto de teste.

## 4.2 O Banco de Faces

O banco de dados de faces da universidade de Yale foi usado nesse trabalho para testar o desempenho da combinação do método DCT com a máquina de aprendizagem SVM.

Este banco contém 165 arquivos no formato gif (Graphic Interchange Format) contendo imagens com dimensões 320x243 de faces em preto e branco de 15 pessoas em 11 situações diferentes. A figura 4.1 apresenta um exemplo de uma pessoa no banco de faces de Yale.



Figura 4.1: Fotos de uma pessoa no banco de Yale, uma foto com luz central, com óculos, feliz, luz a esquerda, sem óculos, normal, luz a direita, triste, sonolento, surpreso e piscando um olho. Imagens extraídas do próprio banco de dados de faces de Yale.

Todas as imagens deste banco já apresentam normalização geométrica, portanto não foi necessário aplicar nenhum método para isso. Também não foi aplicado nenhum método de normalização de iluminação, apesar de estes terem sido discutidos anteriormente.

## 4.3 Usando as Faces na SVM-Multiclass

Para transformar as imagens em vetores e gerar os módulos de treinamento e

classificação para que pudessem ser usados pela SVM foi desenvolvido um programa em Delphi 5.

Primeiro as imagens foram convertidas do formato gif para o formato bmp (bitmap). Esta conversão foi feita através de um conversor externo, e não pelo programa desenvolvido.

A seguir usou-se o software desenvolvido para ler cada arquivo do banco, e para cada arquivo, aplicar o método DCT na imagem contida neste arquivo, gerando um vetor de 64 características (coeficientes). Este vetor então é normalizado e inserido como uma nova linha no arquivo que servirá de entrada para o módulo de aprendizagem da implementação da SVM. A classe desta linha é definida de acordo com a pessoa na foto, já que os arquivos de entrada estão numerados por pessoa.

## 4.4 Experimentos Realizados

Para avaliar o método de reconhecimento facial desenvolvido foram realizados experimentos variando-se os conjuntos de treinamento e teste e o coeficiente responsável por limitar os multiplicadores de Lagrange na SVM.

Como cada pessoa é representada no banco de dados de faces por 11 fotos, foram gerados 11 conjuntos de treinamento e 11 conjuntos de teste distintos. Os conjuntos de testes foram gerados a partir da retirada de uma única foto de cada pessoa, do total de 11 fotos. As outras 10 fotos restantes de cada indivíduo formaram o conjunto de aprendizado. Deste modo, em cada experimento o conjunto de aprendizado possui 150 vetores e o conjunto de testes possui 15 vetores.

As imagens foram comprimidas com o método da Transformada Co-seno Discreta, gerando vetores de 64 posições. Como as imagens possuem dimensões 320x243, a taxa de compressão foi de 1215:1.

Além disso, também foram testadas variações no valor do limite dos coeficientes de Lagrange da equação de aprendizado da SVM. Para escolher quais valores deste coeficiente seriam usados nos experimentos foi feito um teste prévio usando apenas os conjuntos de aprendizado e teste formados a partir da retirada da primeira imagem de cada indivíduo. O resultado desse pré-teste esta na tabela 4.1.

Este fator C que limita o valor dos multiplicadores de Lagrange é importante pois

ele é responsável pelo *tradeoff* entre o erro de treinamento e a maximização da margem de separação entre as classes. Aumentando o erro de treinamento ignoramos vetores que possam estar muito distantes do restante de sua classe, e assim podemos aumentar a área de separação entre as classes. Diminuindo o erro de treinamento diminuimos a área de separação entre as classes, e tornamos a superfície de separação mais flexível, o que implica em diminuir a capacidade de generalização da máquina. Aumentar de mais este erro porém torna as classificações feitas pela máquina não confiáveis, por isso é preciso encontrar um valor de equilíbrio, onde as classes estejam separadas da melhor forma possível.

Tabela 4.1: Resultado de experimento com diversos valores de C (limite dos multiplicadores de Lagrange)

<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,001	60	9
0,008	73,333	11
0,009	73,333	11
0,01	80	12
0,02	73,333	11
0,03	73,333	11
0,04	80	12
0,05	73,333	11
0,07	60	9
0,1	66,666	10
1,0	66,666	10

Observando a tabela 4.1 pode-se perceber que os melhores resultados obtidos foram para os valores de  $C = 0,01$  e  $C = 0,04$ . Portanto os testes com os demais conjuntos de aprendizagem e teste foram feitos com os valores de  $C$  0,01; 0,02; 0,03 e 0,04.

## 5 RESULTADOS

Os resultados são apresentados a seguir em 11 tabelas, uma para cada par de conjuntos de aprendizagem e teste gerados, e apresentam o número de pessoas identificadas corretamente pelo método desenvolvido para cada um dos quatro valores usados como limites dos multiplicadores de Lagrange testados Além disso, a porcentagem de acerto (acurácia) do método é calculada para cada experimento. A análise dos resultados é realizada ao final do capítulo.

Tabela 5.1: Resultados do conjunto de aprendizagem e teste 1

<b>Usando a foto 1 como teste – Iluminação frontal</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	80	12
0,02	73,333	11
0,03	73,333	11
0,04	80	12

Tabela 5.2: Resultados do conjunto de aprendizagem e teste 2

<b>Usando a foto 2 como teste – Com óculos</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	73,333	11
0,02	80	12
0,03	80	12
0,04	80	12

Tabela 5.3: Resultados do conjunto de aprendizagem e teste 3

<b>Usando a foto 3 como teste - Feliz</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	93,333	14
0,02	93,333	14
0,03	100	15
0,04	93,333	14

Tabela 5.4: Resultados do conjunto de aprendizagem e teste 4

<b>Usando a foto 4 como teste – Luz a esquerda</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	20	3
0,02	13,333	2
0,03	13,333	2
0,04	13,333	2

Tabela 5.5: Resultados do conjunto de aprendizagem e teste 5

<b>Usando a foto 5 como teste – Sem óculos</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	93,333	14
0,02	93,333	14
0,03	93,333	14
0,04	93,333	14

Tabela 5.6: Resultados do conjunto de aprendizagem e teste 6

<b>Usando a foto 6 como teste - Normal</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	100	15
0,02	100	15
0,03	100	15
0,04	100	15

Tabela 5.7: Resultados do conjunto de aprendizagem e teste 7

<b>Usando a foto 7 como teste – Luz a direita</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	0	0
0,02	13,333	2
0,03	6,666	1
0,04	0	0

Tabela 5.8: Resultados do conjunto de aprendizagem e teste 8

<b>Usando a foto 8 como teste - Triste</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	86,666	13
0,02	93,333	14
0,03	93,333	14
0,04	93,333	14

Tabela 5.9: Resultados do conjunto de aprendizagem e teste 9

<b>Usando a foto 9 como teste - Sonolento</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	86,666	13
0,02	86,666	13
0,03	93,333	14
0,04	93,333	14

Tabela 5.10: Resultados do conjunto de aprendizagem e teste 10

<b>Usando a foto 10 como teste - Surpreso</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	86,666	13
0,02	80	12
0,03	73,333	11
0,04	73,333	11

Tabela 5.11: Resultados do conjunto de aprendizagem e teste 11

<b>Usando a foto 11 como teste – Piscando um olho</b>		
<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 15)</b>
0,01	80	12
0,02	80	12
0,03	80	12
0,04	80	12



Tabela 5.12: Resultado geral por valor de C

<b>C</b>	<b>% acertos</b>	<b>Pessoas identificadas corretamente (em 165)</b>
0,01	80	132
0,02	73,333	121
0,03	73,333	121
0,04	72,727	120

A tabela 5.12 foi montada com base nos dados obtidos das tabelas 5.1 a 5.11 e mostra que utilizando o limite 0,01 para os multiplicadores de Lagrange a Support Vector Machine obteve melhores resultados.

Uma observação interessante é que como não foi feito nenhum tratamento de normalização de iluminação, os piores resultados obtidos foram justamente ao tentar classificar as fotos onde a iluminação é lateral, como observado nas tabelas 5.4 e 5.7.

No geral a acurácia do método de reconhecimento facial desenvolvido foi de 74,848% para o banco de dados de faces de Yale. É importante destacar que este valor de acurácia só pode ser comparado com a acurácia obtida de outros métodos para este mesmo banco de imagens.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

O método proposto para reconhecimento facial baseado na Transforma Co-seno Discreta e na Máquina de Vetores de Suporte apresentou-se satisfatório.

A acurácia de 74,848% é um bom resultado, visto que não foi feito nenhum tratamento de iluminação nas imagens testadas. Se ignorarmos os testes com as fotos quatro e sete obtemos um valor de acurácia muito maior, de 89,258% no geral, e de 95.555% se considerarmos apenas os dados obtidos com  $C = 0,01$ .

Com base nesta observação, fica evidente que um trabalho que introduza o tratamento de iluminação no método de reconhecimento facial descrito poderá melhorar em muito o desempenho deste método.

Outro trabalho que seria importante para permitir uma avaliação melhor do método de reconhecimento descrito nesse trabalho seria aplicar esse método em outras bases de faces, afim de comparar a acurácia obtida com outros trabalhos realizados sobre estas mesmas bases de dados.

Comparações com outras técnicas holísticas de extração de características de imagens como KLT (Karhunen-Loève Transform) e Gabor Wavelets também são importantes como trabalhos futuros com o intuito de comprovar a eficiência do DCT.

## 7 REFERENCIAS BILIOGRÁFICAS

ABATE Andrea F., NAPPI Michele, RICCIO Daniel, SABATINO Gabriele (2007). 2D and 3D face recognition: A survey. Pattern Recognition Letters Vol. 28.

BENNETT Kristin P. e CAMPBELL Colin (2000). Support Vector Machines: Hype or Hallelujah? ACM SIGKDD Explorations Newsletter. Vol. 2.

BRUNELLI Roberto e POGGIO Tomaso (1993). Face Recognition: Features versus Templates. Transactions on Pattern Analysis and Machine Intelligence. Vol. 15 Num. 10.

CARVALHO, Bernardo Penna Resende de (2005), Novas Estratégias para Detecção Automática de Vetores de Suporte em Least Square Support Vector Machines. Trabalho de pós-graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais.

ER Meng Jo, CHEN Weilong e WU Shiqian (2005). High-Speed Face Recognition Based on Discrete Cosine Transform and RBF Neural Networks. Transactions on Neural Networks, Vol. 16 Num. 3

GROSS R. Matthews, BAKER S.(2002) apud ABATE (2007). Eigen light-fields and face recognition across pose. In Proc. Fifth IEEE Internat. Conf. On Automatic Face and Gesture Recognition,

HAFED Ziad M., LEVINE Martin D. (2001), Face Recognition Using Discrete Cosine Transform. International Journal of Computer Vision Vol.43 Num. 3.

HAYKIN, Simon (1999), Redes Neuras Princípios e Prática 2ª Edição. Editora Prentice Hall, Inc.

KIRBY M. e SIROVICH L. (1990). Application of the Karhunen-Loève Procedure for the Characterization of Human Faces. Transactions on Pattern Analysis and Machine

Intelligence. Vol. 12 Num. 1.

SAKAI T., NAGAO M. e FUJIBAYASHI S. (1969) apud HAFED (2001). Line extraction and pattern recognition in a photograph. *Patt. Recog*, 1:233-248

SEKULER R. e BLAKE R. (1994) apud HAFED (2001). *Perception*. 3ª edição. Editora McGraw-Hill. New York NY

TURK Matthew A. e PENTLAND Alex P. (1991). *Face Recognition Using Eigenfaces*.

VAPNIK V. N. (1982) apud CARVALHO (2005). Estimation of dependences based on empirical data. Editora Springer-Verlag

VAPNIK, Vladimir N. (1999), *An Overview of Statistical Learning Theory*, *IEEE Transactions on Neural Networks*, vol 10 n° 5.

ZHAO W., CHELLAPPA R., PHILLIPS P. J., ROSENFELD A., *Face Recognition: A Literature Survey*. *ACM Computing Surveys*, Vol 35, Num. 4.