



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

**Classificação de pedestres baseada em
reconstrução modular**

Guilherme Vasconcelos de Carvalho

Trabalho de Graduação

Recife
14 de dezembro de 2010

Universidade Federal de Pernambuco
Centro de Informática

Guilherme Vasconcelos de Carvalho

Classificação de pedestres baseada em reconstrução modular

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *George Darmiton da Cunha Cavalcanti*
Co-orientador: *Tsang Ing Ren*

Recife
14 de dezembro de 2010

Agradecimentos

Meus agradecimentos por esse trabalho vão, inicialmente, para toda a minha família pelo apoio, pela compreensão nas minhas ausências e pelos exemplos que eles representam. Sem isso, este trabalho não poderia ser realizado.

Em segundo lugar, ao meu orientador, que fez juz ao título e cuja ajuda e calma foram de fundamental importância quando as coisas não pareciam muito boas.

Aos meus amigos e colegas, que fizeram os quatro anos de curso muito mais fáceis de suportar (exceto nos finais de período, quando não tinha jeito mesmo). Em especial aos que tiveram uma convivência mais próxima: Bruno, Camila, Emanuel, Lailson e Luis.

Ainda no Cin, agradeço por todos os amigos que pude fazer, por também terem facilitado esta caminhada. A todos que foram meus monitorandos, e aos que não foram, mas se chegaram mesmo assim. Em especial a Amora, Débora, Livia e Mariana, pela amizade e carinho que me foi dado.

E por último, porém não menos importante, aos meus amigos que, apesar de não entenderem como ficar no computador o dia todo pode ser trabalhoso, não me deixaram abandonar de vez a minha vida social: Augusto, Andréa, André, Amélia, Neto, Maia e Regina. A Alicia, que me acompanhou por quase todos estes anos e por quem eu tenho a mais alta admiração, eu não poderia deixar passar uma segunda oportunidade de agradecer.

Igziabeher!

Apressa-te a viver bem e pensa que cada dia é, por si só, uma vida
—SÊNECA

Resumo

Detecção de pedestres é uma tarefa que tem aplicações nas mais diversas áreas. Suas utilizações variam de sistemas de auxílio ao motorista, onde tem o objetivo de diminuir as taxas de atropelamento, a sistemas de segurança, onde visa detectar a presença de humanos em locais restritos, detectar ações suspeitas, entre outras aplicações.

Este trabalho analisa a viabilidade e propõe a construção de um classificador que utiliza reconstrução de componentes de um pedestre para a classificação. Os resultados obtidos mostram que é possível a criação de um classificador que utiliza apenas informações do torso e das pernas de um pedestre, obtendo assim uma classificação mais rápida.

Palavras-chave: detecção de pedestres; PCA; PCA modular; reconhecimento de padrões; detecção de objetos

Abstract

Pedestrian detection is a task with a great range of applications. Its uses vary from automated driver assistance systems (ADAS), aiming to decrease the number of accidents involving pedestrians, to security systems, where it's used to detect the presence of humans in areas with restricted access, activity analysis, among others.

This paper analyses the feasibility and proposes the construction of a classifier that uses the reconstruction error from pedestrian components for classification. Results obtained show that it is possible to create a classifier using only information from the torso and the legs of a pedestrian thereby obtaining a faster classification.

Keywords: pedestrian detection; PCA; Modular PCA; pattern recognition; object detection

Sumário

1	Introdução	1
2	Conceitos básicos	3
2.1	Autovalores e autovetores	3
2.2	Covariância	4
2.3	PCA	5
2.3.1	Cálculo dos PCs	5
2.3.2	História do uso de PCA em visão computacional	6
2.4	PCA Modular	7
2.4.1	Formulação	7
2.4.2	Discussão	8
2.5	Algoritmos de detecção de bordas	8
2.5.1	Convolução	9
2.5.2	Discussão	10
2.6	Algoritmos Genéticos	10
2.6.1	Conceitos biológicos	10
2.6.2	Formulação	11
2.6.3	Discussão	12
2.6.4	Exemplo	12
3	Classificador	15
3.1	Classificação baseada em reconstrução	15
3.1.1	Treinamento	16
3.1.2	O classificador	17
3.1.3	Discussão	17
3.2	Classificador baseado em reconstrução modular	19
3.2.1	Utilização de PCA Modular	19
3.2.2	Treinamento	20
3.2.3	O classificador	21
3.2.4	Discussão	22
4	Experimentos	23
4.1	Base de dados	23
4.1.1	Imagens de pedestres	23
4.1.2	Imagens de não pedestres	24
4.1.3	Conjuntos de treinamento e teste	24

4.2	Utilização do Algoritmo Genético	24
4.3	Avaliação dos Classificadores	24
4.4	Análise das hipóteses	25
4.4.1	Equivalência dos classificadores	26
4.4.2	Redundância das imagens tons de cinza e de borda	28
4.4.3	Classificadores de partes	29
4.4.4	Classificação modular com pesos	30
5	Conclusão	33
A	Formulação dos classificadores parciais	35
A.1	Classificador cabeça/torso	35
A.1.1	Treinamento	35
A.1.2	O classificador	35
A.2	Classificador cabeça/pernas	36
A.2.1	Treinamento	36
A.2.2	O classificador	37
A.3	Classificador torso/pernas	38
A.3.1	Treinamento	38
A.3.2	O classificador	38

Lista de Figuras

2.1	Reconstrução baseada em PCA modular	8
2.2	Aplicação do operador de Sobel	11
3.1	Erro de reconstrução	15
3.2	Imagem de bordas de um pedestre	16
3.3	Comparação da AUC do classificador com e sem pesos	18
3.4	Divisão de uma imagem de pedestre para a classificação baseada em PCA Modular	20
4.1	AUC vs k para o classificador original	26
4.2	AUC vs k: borda x tons de cinza x ambas	28
4.3	AUC vs k: classificadores parciais	29
4.4	Curva ROC classificador baseado em PCA modular com e sem pesos	32

Lista de Tabelas

3.1	Tempo de detecção médio (ms)	19
4.1	AUC vs k para classificadores com imagens em tons de cinza	27
4.2	AUC vs k para classificadores com pesos	27
4.3	AUC vs k para classificadores com imagens de borda	27
4.4	Verdadeiros positivos (TP) para classificadores total e torso/pernas	30
4.5	Tempo de detecção médio para reconstrução modular (ms)	30
4.6	AUCs para classificadores parciais sem e com a utilização pesos	31

Introdução

Técnicas de detecção de pedestre tem sido amplamente utilizadas e pesquisadas na área de visão computacional [Kan07]. Suas aplicações vão desde detecção em imagens até sistemas de auxílio ao motorista (ADAS, do inglês *Advanced Driver Assistance Systems*). Apesar de ser um caso especial da detecção de objetos, a detecção de pedestres apresenta certas características que tornam a tarefa mais difícil.

Pedestres podem aparecer sob diferentes condições de iluminação e com cenários de fundo complexos, como uma cidade. Além disso humanos são altamente articulados e podem assumir diferentes posições. Eles têm uma grande variedade de tamanho, podem utilizar roupas que alteram sua textura e não raramente são vistos carregando objetos oclusivos, como mochilas ou carrinhos de bebê, que alteram consideravelmente suas silhuetas.

A tarefa de detecção de pedestres está intimamente ligada ao problema de vídeo segurança. Seu uso pode variar desde a identificação da presença de um pedestre em uma região com acesso limitado até a análise de atividades. Independente do uso de vídeo-segurança, a utilização de técnicas de detecção de pedestres é imprescindível.

No contexto de ADAS, há ainda mais fatores complicantes, pois além do cenário de fundo e da variabilidade do pedestre, a detecção tem que ser feita em tempo real e o número de falsos positivos deve ser baixo [Ger09]. Há ainda a complicação da câmera e o pedestre estarem em movimento, o que faz com que os pedestres tenham que ser detectados em diferentes ângulos de visão

As técnicas para detecção de pedestres podem ser baseadas em uma única imagem ou em uma sequência de imagens. Os detectores que se baseiam em uma única imagem geralmente tentam modelar um pedestre, baseando-se em sua fisionomia completa [MBF09], ou baseando-se em detecção das partes de um pedestre, como pernas, braços e cabeça [MSZ04] [Moh01] [Sha04]. Detectores baseados em sequências de imagens podem utilizar informação do movimento [VJS05] [Lee02] [Cut00], que é bastante característico de humanos [DTS06]. Classificadores baseados em movimento e forma geralmente obtêm resultados melhores do que os apenas baseados em forma [DTS06]. Além disso, esses classificadores podem ser utilizados para a análise de atividades [LCY⁺09][Cut00], com resultados positivos.

Este trabalho tem como objetivo analisar a viabilidade de construir um sistema de detecção de pedestres em imagens estáticas, que baseie-se em reconstrução de componentes inspirado na utilização de Gottumakal da técnica de PCA Modular [GA04].

O classificador de imagens proposto baseia-se no criado por Malagón-Borja e Fuentes em [MBF09], pois utiliza os erros de reconstrução obtidos a partir da projeção de

uma imagem em um espaço construído pela aplicação do PCA. Porém o classificador proposto usa informações de partes do pedestre, ao invés de tentar modelar todo o corpo do pedestre.

O resto do trabalho se divide como segue: o Capítulo 2 expõe os conceitos básicos para o entendimento da técnica utilizada na classificação; o Capítulo 3 analisa o classificador de Malagón-Borja e Fuentes [MBF09] e descreve o classificador proposto neste trabalho; o Capítulo 4 demonstra e analisa os resultados obtidos pelo classificador e compara-os com os resultados do classificador original.

Conceitos básicos

Para que seja possível entender o classificador proposto por este trabalho, faz-se necessária a explicação de conceitos de várias áreas de conhecimento. Este capítulo apresenta tais conceitos na seguinte ordem: o conceito de autovalores e autovetores é apresentado na Seção 2.1; os conceitos estatísticos de covariância e PCA são apresentados nas Seções 2.2 e 2.3, respectivamente; os conceitos da área de Visão Computacional, PCA modular e algoritmos de detecção de bordas são apresentados nas Seções 2.4 e 2.5; e, finalmente, o conceito de Algoritmos Genéticos é apresentado na Seção 2.6.

2.1 Autovalores e autovetores

Autovalores e autovetores são propriedades de matrizes quadradas que são calculadas para obter informações quanto à sua estrutura [Mar88]. Tais propriedades tem aplicação em diversas áreas como mecânica quântica, onde são utilizadas para avaliar o estado de um sistema quântico; análise de oscilações em sistemas vibratórios, onde são utilizadas para determinar a frequência natural de vibração do sistema; e em álgebra linear, onde são utilizadas para o estudo de transformações lineares. Sua definição vem do estudo de transformações lineares, porém os conceitos podem ser aplicados a qualquer matriz, desde que a mesma seja quadrada.

A definição de autovalor e autovetor é dada pela equação:

$$Ax = \lambda x \tag{2.1}$$

Dada uma matriz A a partir da qual se deseja calcular os autovetores e autovalores, x é o autovetor e λ é um escalar associado a esse autovetor, chamado de autovalor. Como pode ser deduzido a partir da equação, os autovetores são vetores do espaço original que não tem sua direção alterada pela transformação linear representada por A , sendo apenas escalados pelo fator λ .

É importante observar que nem toda matriz quadrada pode ser decomposta em autovetores e autovalores, e que é possível a existência de autovalores imaginários. Além disso, é importante explicitar as seguintes propriedades de autovetores:

- O vetor nulo, constituído apenas de zeros, não é um autovetor, pois ele valida a equação independente de quem seja a matriz A e o autovalor λ .

- Todos os autovetores de uma matriz são linearmente independentes entre si, o que os torna bons candidatos para serem base de um espaço.

Para o cálculo dos autovetores e autovalores de uma matriz quadrada A , podemos reescrever a definição como:

$$Ax - \lambda Ix = (A - \lambda I)x = 0 \quad (2.2)$$

Se o determinante de $(A - \lambda I)$ for diferente de 0, sabemos que este sistema tem uma única solução, o vetor nulo. Como queremos os autovetores de A , ou seja, solução não nulas, devemos forçar que o determinante de $(A - \lambda I)$ seja igual a 0. A expansão da equação do determinante nos dá um polinômio em λ , que é conhecido como o polinômio característico de A [Bol86]. A resolução da equação característica tem como resultado os autovalores de A , e a substituição de cada autovalor em 2.1 resulta nos autovetores associados a cada autovalor.

2.2 Covariância

A covariância entre duas variáveis aleatórias é uma medida que indica a relação de dependência entre essas variáveis, ou seja, o quanto a alteração do valor de uma variável influi no valor da outra. Duas propriedades da covariância são: quanto maior a covariância entre duas variáveis, mais elas são correlacionadas; variáveis independentes tem o valor da covariância nulo, porém a covariância zero não implica que as variáveis são independentes.

Para o cálculo da covariância, define-se o valor esperado de uma variável discreta X como:

$$E(X) = \sum_{i=1}^n x_i p(x_i), \quad (2.3)$$

onde $p(x_i)$ é a probabilidade da variável X assumir o valor x_i .

A covariância entre duas variáveis aleatórias, X e Y , pode ser calculada através da equação

$$\sigma_{X,Y} = E(XY) - E(X)E(Y). \quad (2.4)$$

A matriz de covariância é uma ferramenta útil para a visualização das covariâncias de um grande número de variáveis. Considerando um vetor de variáveis aleatórias \mathbf{u} , a matriz de covariância é definida como uma matriz onde cada elemento (i, j) representa o valor da covariância entre u_i e u_j . É importante ressaltar que a matriz de covariância é quadrada, que os elementos da diagonal principal (i, i) , contêm o valor da variância de u_i e que a matriz de covariância é uma matriz simétrica, pois a igualdade $\sigma_{X,Y} = \sigma_{Y,X}$ é verdadeira.

2.3 PCA

As bases para a técnica conhecida como PCA (*Principal Component Analysis*) foram descritas por Karl Pearson [Pea01] em 1901, com o intuito de representar de forma ótima dados provenientes de estudos de sistemas e facilitar a análise de dados multivariados. Porém, a sua formulação atual foi feita por Hotelling [Hot33], onde o nome PCA foi citado pela primeira vez.

Desde então, a utilização de PCA foi expandida para mais áreas, além do estudo de sistemas, como: extração de informações e análise estrutural de um conjunto de dados; e compressão de dados. Para tais fins a técnica visa, a partir de um conjunto de observações com variáveis possivelmente redundantes, encontrar um conjunto de novas variáveis, chamadas componentes principais (*principal components*, PCs, em inglês), que sejam minimamente correlacionadas e que mantenham o máximo possível da variabilidade do conjunto inicial.

Para obter tal conjunto de novas variáveis o PCA baseia-se em quatro suposições básicas [Sh109] que visam explicar o funcionamento do PCA e provêm um entendimento básico para saber quando a aplicação de PCA pode gerar bons resultados e quando isto pode não acontecer. As quatro suposições são:

1. Os dados sendo analisados devem ser lineares para que seja possível a troca de base aplicando conceitos de álgebra linear;
2. A média e a variância devem ser medidas suficientes para representar a distribuição probabilística dos dados. Essa suposição garante que a matriz de covariância caracteriza completamente a redundância das variáveis iniciais;
3. Altas variâncias representam informações relevantes, pois a escolha dos componentes principais é baseada na variância;
4. Os componentes principais são ortogonais de modo a garantir que a solução possa ser encontrada utilizando álgebra linear.

Existem várias maneiras para calcular os PCs de um conjunto de dados: utilizando a matriz de covariância dos dados, a matriz de correlação, ou utilizando uma técnica conhecida como SVD (*singular value decomposition*) [Sh109]. Neste trabalho utilizamos o cálculo baseado na matriz de covariância.

2.3.1 Cálculo dos PCs

Os componentes principais de um conjunto com n de observações, cada uma com m características, onde cada observação é representada por um vetor-coluna x_i , é calculado da seguinte maneira:

- A média do conjunto é calculada:

$$\bar{\mu} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.5)$$

- Todas as mostras são centralizadas para que seja possível encontrar os componentes principais que minimizem o erro quadrático médio:

$$\bar{x}_i = x_i - \bar{\mu} \quad (2.6)$$

- Calcula-se a matriz de covariância do conjunto através da fórmula:

$$C = \sum_{i=1}^n \bar{x}_i \bar{x}_i^T \quad (2.7)$$

- Os autovetores e autovalores são computados a partir da matriz de covariância e os autovetores são ordenados pelos respectivos autovalores, de modo que o primeiro componente principal seja o autovetor com o maior autovalor associado, o segundo componente principal tenha o segundo maior autovalor e assim sucessivamente.

Na formulação inicial do PCA, todos os autovetores eram utilizados como bases para um novo espaço projetivo, porém a análise dos autovalores (quantidade de informação presente em cada nova dimensão) mostrou que geralmente existe um $k < n$, tal que a maior parte da informação está contida nos k primeiros componentes principais. Esta observação levou à aplicação de PCA como técnica de redução de dimensionalidade.

Para utilizar o PCA como técnica de redução de dimensionalidade, devemos ordenar os k primeiros componentes principais como vetores colunas para obter uma matriz de projeção. Essa matriz projeta as amostras iniciais de um espaço n -dimensional em um espaço k -dimensional, onde a perda de informação pode ser ajustada de acordo com o valor do parâmetro k .

2.3.2 História do uso de PCA em visão computacional

As primeiras utilizações de PCA aplicadas à área de visão computacional foram descritas inicialmente por Sirovich e Kirby [SK87] e depois por Turk e Pentland [TP91]. No seu artigo [TP91], Turk e Pentland propõem uma técnica para o reconhecimento de faces utilizando um espaço projetivo chamado face-space. A técnica descrita conseguiu resultados significantes, e seu baixo custo computacional possibilitou a criação de um sistema de detecção e reconhecimento de faces em tempo real.

Para a construção do face-space, a técnica extrai do conjunto de treinamento uma determinada quantidade de PCs, e utiliza os PCs como bases para o espaço. Para reconhecer uma nova face, o sistema a projeta no face-space, e utilizando uma métrica de distância, classifica a nova face com o rótulo da face que esteja mais próxima dela nesse novo espaço.

A inovação proposta por Turk e Pentland consiste em perceber que numa imagem de dimensões $M \times N$, nem todas as características são relevantes e que apenas um subconjunto das características, geralmente muito menor do que MN , pode ser utilizado sem que haja uma perda significativa de informação. Graças a esta diminuição

no número de características sendo analisadas, e sua consecutiva diminuição no custo computacional, possibilitou a criação de um sistema em tempo real.

A importância dos artigos de Sirovich e Kirby [SK87] e Turk e Pentland [TP91] é tamanha que estes artigos estão entre os mais citados na área de Visão Computacional.

2.4 PCA Modular

A técnica conhecida como PCA modular foi inicialmente proposta por Rajkiran Gottumukkal e Vijayan K. Asari [GA04]. O uso para o qual foi concebida visava melhorar a classificação baseada em eigenfaces nos casos em que havia uma grande variação angular para a detecção de uma face. A inovação do PCA modular consistia em dividir as imagens de treinamento e teste e extrair os componentes principais individualmente de cada uma dessas partes.

2.4.1 Formulação

No método do PCA modular, cada imagem é dividida em N imagens menores. Todas as N sub-imagens devem ter o mesmo tamanho, L^2/N , e podem ser definidas matematicamente para uma imagem I_1 como:

$$I_{1j}(m, n) = I_1\left(\frac{L}{\sqrt{N}}(j-1) + m, \frac{L}{\sqrt{N}}(j-1) + n\right) \quad \forall j \quad (2.8)$$

onde j vai de 1 a N e m e n vão de L/\sqrt{N} .

Com as sub-imagens construídas, calculamos sua média através da equação:

$$A = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N I_{ij} \quad (2.9)$$

Para o cálculo do PCA, precisamos da matriz de covariância, calculada pela equação:

$$C = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (I_{ij} - A)(I_{ij} - A)^T \quad (2.10)$$

A partir da matriz de covariância são extraídos M' autovetores. Esses autovetores são utilizados para projetar as imagens de treinamento no eigenspace através da equação:

$$W_{pnjK} = E_K^T(I_{pnj} - A) \quad \forall p, n, j, K \quad (2.11)$$

onde K varia de 1 até M' , n varia de 1 até Γ , onde Γ é o número de imagens por indivíduo e p varia de 1 a P onde P é o número de indivíduos na base de treinamento.

É importante ressaltar que este passo está intrinsecamente ligado ao problema em questão, que é o reconhecimento de face, onde as bases apresentam várias imagens de um mesmo indivíduo com variações de iluminação ou expressão facial. Caso não levássemos em consideração a presença de vários indivíduos sob algumas variações poderíamos reescrever a equação 2.4.1 de forma mais simples como:

$$W_{mjK} = E_K^T(I_{mj} - A) \quad \forall m, j, K \quad (2.12)$$

Ao classificar uma nova imagem, o algoritmo a divide em N partes utilizando a equação 2.4.1 e então calculando seus pesos no eigenspace através da fórmula:

$$W_{testjK} = E_K^T(I_{test} - A) \quad \forall j, K \quad (2.13)$$

e a partir dos pesos obtidos utiliza um classificador simples, como o kNN [CH67], para rotular a imagem de acordo com as regras do classificador.

2.4.2 Discussão

A análise do PCA Modular feita por Gottumukkal mostra que existem situações em que a utilização de informações locais na construção do PCA resulta em melhores classificações, principalmente sob variações de iluminação e de expressão facial. Além disso, como pode ser visto na figura 2.1, a reconstrução com PCA modular é mais fidedigna do que a reconstrução por PCA, devido ao fato de as informações do PCA modular serem locais, ao invés de globais, como no caso do PCA normal.



Figura 2.1 Exemplo de reconstruções utilizando PCA e PCA modular. Na figura da esquerda, a reconstrução baseada em PCA; na figura do meio, a imagem original; e na figura da direita, a reconstrução feita com PCA modular e com $N = 4$. Na base de treinamento utilizada para o cálculo dos pesos havia imagens com variação de iluminação e expressão facial.

2.5 Algoritmos de detecção de bordas

A tarefa de detectar as bordas de uma imagem é de extrema importância para o processamento de imagens. Tem aplicações principalmente na tarefa de segmentação automatizada de imagens, que consiste em particionar uma imagem em diferentes regiões

que tenham algum significado para a aplicação. Além disso a detecção de bordas pode ser utilizada para aperfeiçoamento de imagens e restauração, compressão e para a tarefa reconhecimento de objetos [ZT98].

Uma das técnicas mais utilizadas e simples de entender, e a utilizada neste trabalho, para a detecção de borda em imagens foi o operador de Sobel [GWE03]. O operador foi inicialmente proposto por Irwin Sobel em um artigo não publicado de 1967. Sua formulação matemática consiste em dois filtros espaciais que calculam uma aproximação do gradiente da imagem tanto na direção horizontal (S_x) quanto na vertical (S_y).

$$S_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} \quad (2.14)$$

$$S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} \quad (2.15)$$

Os dois filtros podem ser combinados para gerar uma nova imagem, utilizando a aproximação $G = \sqrt{G_x^2 + G_y^2}$, cujo efeito é análogo ao de computarmos uma das seguintes somas: $G_x^2 + G_y^2$ ou $|G_x| + |G_y|$. A última soma tem a vantagem de ser computacionalmente mais eficiente. Ambas as aproximações são equivalentes pois têm o valor nulo em áreas de intensidade constante e seus valores são proporcionais à mudança de intensidade nas áreas sendo analisadas.

2.5.1 Convolução

Os operadores são aplicados através de sua convolução pela imagem da qual se deseja obter os contornos. A convolução é um operador matemático que pode ser definido como a quantidade de sobreposição que existe ao passarmos uma função sobre outra. Na área de processamento de imagens o processo de convolução pode ser explicado mais simplesmente como apenas "passar" um filtro, como S_x , com o centro sobre cada pixel da imagem, calculando um novo valor para este pixel baseado nos pesos definidos no filtro.

Um exemplo da aplicação da convolução utilizando o filtro S_x , sobre a matriz A definida abaixo:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,n} \end{pmatrix} \quad (2.16)$$

Resultaria no seguinte cálculo para o novo valor da posição $a_{2,2}$:

$$a'_{2,2} = -1a_{1,1} + 1a_{1,3} - 1a_{2,1} + 2a_{2,3} - 1a_{1,1} + 1a_{1,3} \quad (2.17)$$

O processo de convolução continuaria alinhando o centro do filtro com o pixel $a_{2,3}$ e calculando o seu novo valor, seguindo para o pixel $a_{2,4}$ e assim sucessivamente até chegar ao pixel $a_{m,n}$.

2.5.2 Discussão

O operador de Sobel calcula uma aproximação do gradiente da imagem, gerando altos valores em pixels onde há uma grande variação de intensidade em sua vizinhança. Esta grande variação pode ser vista como um indício de que pode existir uma borda em tal posição e indica também a direção da mudança de intensidade, que pode, em algumas aplicações, indicar se o filtro entrou ou saiu da área interna de um objeto.

A aplicação dos operadores S_x e S_y separadamente, além de sua combinação podem ser vistas na figura 2.2.

2.6 Algoritmos Genéticos

Algoritmos Genéticos foram inventados por John Holland [Hol92] durante a década de 1960, na Universidade de Cambridge, com o intuito de simular e estudar sistematicamente o processo de evolução e adaptação que ocorre na natureza. Apesar do intuito inicial, o algoritmo teve grande uso na área de otimização de funções.

Por ter sido grandemente baseado na natureza, é necessária a explicação de alguns termos da biologia para entendê-lo completamente.

2.6.1 Conceitos biológicos

Todos os seres vivos possuem cromossomos, que são sequências de DNA que podem ser vistas como um guia para a criação e desenvolvimento do organismo. Os cromossomos são constituídos de genes, que são instâncias de um alelo, e representam uma determinada característica do indivíduo, como o tipo sanguíneo. Um alelo é o conjunto de valores possíveis para um gene. Para o exemplo do tipo sanguíneo os possíveis seriam A, B ou O.

No ato da reprodução celular, cromossomos podem ser combinados (operação conhecida como *crossover*) gerando assim novos indivíduos com partes do DNA paterno e do materno. Além disso, durante a reprodução, a prole pode sofrer uma mutação genética, que é uma alteração aleatória no seu DNA.

Além da genética, outro campo da biologia utilizado para a formulação de algoritmos genéticos é o de evolução, mais especificamente o conceito de seleção natural, que dita que os indivíduos mais aptos tem uma maior probabilidade de se reproduzirem.

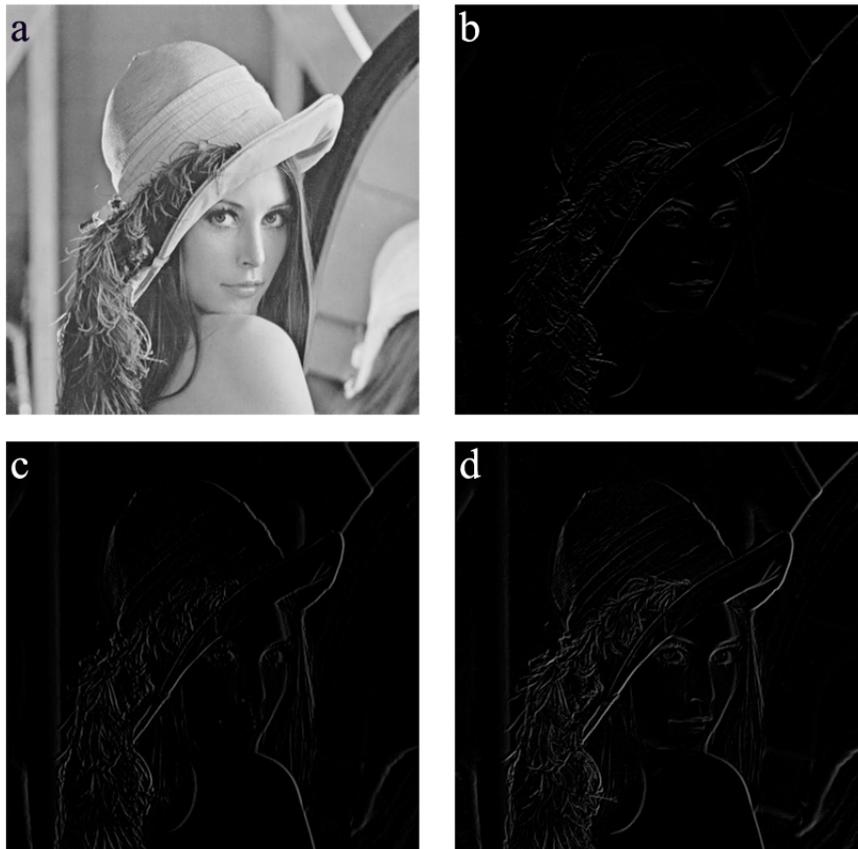


Figura 2.2 Exemplo de aplicação do operador de Sobel na imagem de Lena. A figura (a) mostra a imagem original; a figura (b) mostra a aplicação do operador S_x em (a); a figura (c) mostra a aplicação do operador S_y em (a); e a figura (d) mostra a combinação de (b) e (c).

2.6.2 Formulação

Baseado nos conceitos apresentados em 2.6.1, Holland propôs a seguinte formulação para um Algoritmo Genético simples:

1. Comece com uma população inicial de tamanho n gerada aleatoriamente;
2. Calcule a aptidão de cada indivíduo, representado por seu cromossomo, na população;
3. Repita os passos a seguir até que n descendentes tenham sido criados:
 - (a) Selecione um par de cromossomos pai da população atual. A probabilidade de seleção é diretamente proporcional a aptidão de um indivíduo. (Um mesmo indivíduo pode ser escolhido para ser pai mais de uma vez);

- (b) Com uma probabilidade p_c (probabilidade de *crossover*), faça o cruzamento dos cromossomos numa posição escolhida aleatoriamente. Se não houver cruzamento, crie dois filhos iguais aos indivíduos originais;
 - (c) Altere cada gene da prole com uma probabilidade p_m (probabilidade de mutação).
4. Substitua a população anterior com a gerada;
 5. Repita do passo 2 em diante até que um critério de parada seja alcançado.

Cada iteração do Algoritmo Genético é conhecida como geração. Geralmente os algoritmos genéticos tem seu critério de parada baseado na quantidade de gerações, que variam entre 50 e 500 [Mit96]. O conjunto de gerações do Algoritmo Genético até seu critério de parada é conhecida como *run*, ou execução. No final de uma execução é comum a presença de indivíduos com alta aptidão.

2.6.3 Discussão

Por ser baseado na aleatoriedade da população inicial, é comum reportar os resultados para mais de uma execução do Algoritmo Genético, e medidas como a solução mais apta encontrada e em que geração ela foi descoberta.

Apesar de terem sido inicialmente desenvolvidos para o estudo da evolução e adaptação biológica, os algoritmos genéticos são largamente utilizados para busca de soluções onde o espaço das possíveis soluções é muito grande e otimização de funções. Para esta formulação, cada cromossomo é uma possível solução para o problema e sua função de aptidão pode ser vista como uma quantificação de quão bem a solução resolve o problema.

2.6.4 Exemplo

Um exemplo pode ser mostrado para ilustrar a modelagem dos parâmetros de um Algoritmo Genético (estrutura de um indivíduo e função de aptidão). Supondo que temos um classificador que cuja tarefa seja indicar se uma pessoa ganhará mais do que 50 mil dólares em um ano fiscal, baseado nos dados presentes no conjunto encontrado em [FA10].

Cada instância deste conjunto tem 14 características. Suspeita-se, porém, que apenas algumas características sejam necessárias e deseja-se saber quais são as características que podem produzir melhores resultados. Este processo é denominado de seleção de características.

Um seletor de características baseado em algoritmos genéticos pode ser formulado com os indivíduos sendo representados por cromossomos de 14 alelos, cada alelo sendo representado por um bit. O valor 1 para um alelo indica a presença da característica nessa solução e o valor 0 indica a sua ausência.

Uma função de aptidão poderia ser a contagem dos acertos que o classificador faria com essa configuração. Outra função poderia levar em conta a quantidade de falsos

positivos, verdadeiros positivos, falsos negativos e verdadeiros negativos, de modo a deixar o classificador mais seguro, minimizando o número de falsos positivos, ou mais abrangente, preocupando-se apenas com os verdadeiros positivos em detrimento dos falsos positivos.

CAPÍTULO 3

Classificador

Este capítulo descreve um classificador de pedestres que baseia-se na reconstrução proposto por Malagón-Borja e Fuentes [MBF09], que utiliza PCA de uma maneira inovadora. Descreve também um novo classificador baseado em reconstrução modular. A Seção 3.1 explica o classificador de Malagón-Borja e, expõe algumas melhorias propostas pelo autor em [CMRC10]. Por fim, um novo classificador é proposto na Seção 3.2.

3.1 Classificação baseada em reconstrução

O sistema proposto por Malagón-Borja e Fuentes [MBF09] apresenta um novo uso de PCA para a detecção de objetos. Ao contrário do *eigenfaces*, proposto por Turk e Pentland [TP91], a proposta de Malagón-Borja e Fuentes tira proveito do fato de que uma imagem projetada no espaço de PCA e reprojeta de volta para o seu espaço de origem tem um erro associado. Esse erro se deve ao fato de o PCA utilizar apenas alguns PCs para realizar a projeção, havendo assim perda de informação. A diferença entre a imagem original e uma reconstrução pode ser vista na Figura 3.1, abaixo.

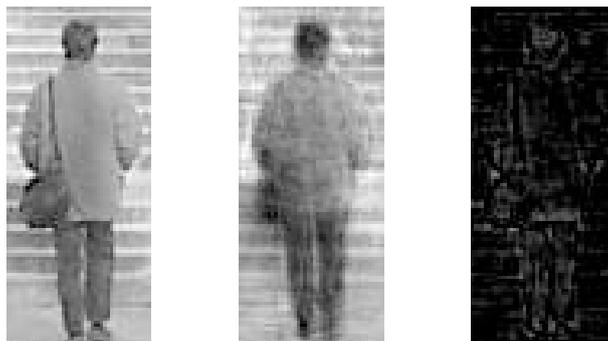


Figura 3.1 À esquerda, a imagem original; no centro, a imagem reconstruída tendo como base um espaço de PCA construído a partir de imagens de pedestres; à direita a diferença entre a imagem original e a reconstruída.

A utilização de PCA nesse sistema é dada a seguir:

- Considere um conjunto C com m imagens, em que cada imagem tem tamanho $w \times h$. Transformamos todas as imagens em vetores coluna v de dimensões wh .

- A partir do conjunto de vetores calculamos o PCA como visto na Seção 2.3.1 e selecionamos k componentes principais para a construção de uma matriz de projeção P , onde cada componente principal é disposto como uma coluna desta matriz.

3.1.1 Treinamento

De acordo com o sistema original de Malagón-Borja e Fuentes, a fase de treinamento necessita de dois conjuntos de imagens: um conjunto de pedestres e um de não-pedestres. A partir desses dois conjuntos, outros dois são gerados aplicando o operador de Sobel nas imagens dos conjuntos originais, obtendo assim quatro conjuntos: imagens de pedestres em tons de cinza, de não-pedestres em tons de cinza, imagens de borda de pedestres e de não-pedestres. Aplicamos PCA como descrito acima para obter quatro matrizes de projeção, a saber, P_{gp} , P_{gn} , P_{ep} e P_{en} , e suas respectivas médias, μ_{gp} , μ_{gn} , μ_{ep} e μ_{en} .

O problema de detecção de pedestres apresenta um grande número de dificuldades, pois pedestres apresentam uma grande variação de altura e pose, apresentam-se com diferentes roupas, o que muda a sua textura na imagem, e ainda podem estar carregando objetos oclusivos. Com a finalidade de diminuir a variação de textura, tanto do plano de fundo quanto das roupas do pedestre, utiliza-se o algoritmo de detecção de bordas de Sobel. A diminuição de variação que ocorre após a detecção das bordas pode ser vista na Figura 3.2.

Os sub-índices das matrizes de projeção e das médias têm os seguintes significados: g significa que o conjunto utilizado para obtê-los continha apenas imagens em *grayscale* (tons de cinza); e significa que o conjunto utilizado continha apenas imagens de borda (do inglês *edge*); p significa que o conjunto utilizado continha apenas imagens de pedestres; e n significa que o conjunto utilizado continha apenas imagens de não-pedestres.



Figura 3.2 À esquerda, a imagem original; à direita a respectiva imagem de bordas. É importante notar a suavização nos detalhes do plano de fundo (como os buracos da rua), dos objetos oclusivos (como a bolsa) e da textura da roupa.

3.1.2 O classificador

Na fase de classificação, dada uma imagem em tons de cinza g realizamos o seguinte procedimento:

1. Obtemos a imagem de bordas e a partir da imagem original g , utilizando o operador de Sobel.
2. Realizamos as quatro reconstruções:

$$(a) \ r_{gp} = P_{gp}P_{gp}^T(g - \mu_{gp}) + \mu_{gp}$$

$$(b) \ r_{ep} = P_{ep}P_{ep}^T(e - \mu_{ep}) + \mu_{ep}$$

$$(c) \ r_{gn} = P_{gn}P_{gn}^T(g - \mu_{gn}) + \mu_{gn}$$

$$(d) \ r_{en} = P_{en}P_{en}^T(e - \mu_{en}) + \mu_{en}$$

3. Calculamos os erros de reconstrução:

$$(a) \ d_{gp} = |r_{gp} - g|$$

$$(b) \ d_{ep} = |r_{ep} - e|$$

$$(c) \ d_{gn} = |r_{gn} - g|$$

$$(d) \ d_{en} = |r_{en} - e|$$

4. A partir dos 4 erros calculamos o erro total utilizando a fórmula

$$d_t = d_{gn} + d_{en} - (d_{gp} + d_{ep}) \quad (3.1)$$

5. Aplicamos a seguinte regra para classificar g :

$$classe(g) = \begin{cases} \text{pedestre} & \text{se } d_t \geq 0 \\ \text{não-pedestre} & \text{se } d_t < 0 \end{cases}$$

3.1.3 Discussão

Ao analisarmos o classificador proposto por Malagón-Borja e Fuentes é óbvia a necessidade de imagens de pedestres para o classificador, afinal precisamos montar um modelo de um pedestre.

A presença, porém, das imagens de não-pedestres parece ser questionável à primeira vista, já que elas não representam nenhum objeto específico. Mas, uma análise da equação 3.1, juntamente com a análise de imagens reconstruídas, mostra que para uma imagem ser classificada como pedestre seu erro de reconstrução baseado nas imagens de pedestres tem que ser menor que na de não-pedestres. Como o conjunto de não-pedestres não modela nenhum objeto, o erro de reconstrução baseado nele é sempre alto. Isto indica que o papel das imagens de não-pedestres é apenas para termos com o que comparar o erro de reconstrução das imagens de pedestres.

A descoberta de que as imagens de não-pedestres servem apenas para contrabalançar o erro de reconstrução dos pedestres, levou Carvalho e Moraes [CMRC10] a propor uma melhoria ao algoritmo original, que visava diferenciar a contribuição de cada erro de reconstrução para o cálculo do erro total, substituindo a equação 3.1 por:

$$d_t = w_{gn}d_{gn} + w_{en}d_{en} - (w_{gp}d_{gp} + w_{ep}d_{ep}) \quad (3.2)$$

Para o cálculo de w_{gn} , w_{en} , w_{gp} e w_{ep} , foi utilizado Algoritmos Genéticos, onde cada indivíduo era constituído dos quatro pesos e o objetivo do Algoritmo Genético era minimizar os erros de classificação utilizando a equação 3.2. A introdução dos pesos à equação 3.1 resultou em melhorias nas taxas de classificação, que podem ser vistas na Figura 3.3.

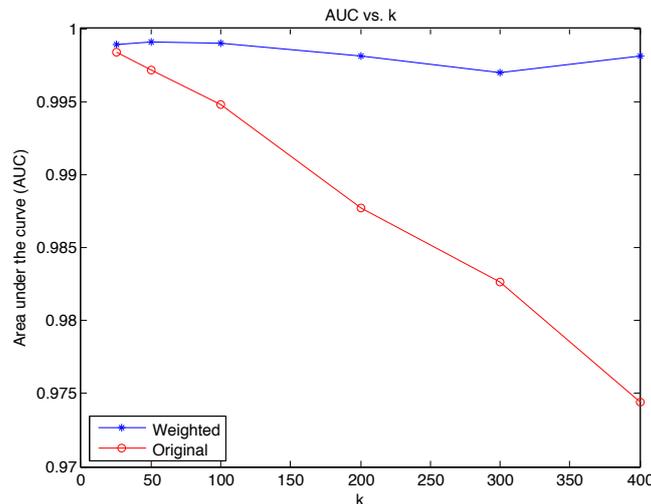


Figura 3.3 Comparação da AUC do classificador com e sem pesos. A AUC do classificador com pesos é melhor e mais constante para vários valores de k .

Outro problema detectado na análise do classificador de Malagón-Borja e Fuentes é uma possível redundância entre imagens em tons de cinza e as respectivas imagens de borda. Como demonstrado na Figura 3.1, as imagens de borda mantêm a silhueta humana e removem da imagem informações que poderiam atuar negativamente no processo de classificação. Também em [CMRC10], foi mostrado que um classificador que desconsidera imagens em tons de cinza e utiliza apenas imagens de borda, tem uma taxa de acerto melhor que o classificador original. Além dessa vantagem, há ainda uma diminuição considerável no tempo de execução já que apenas metade das projeções serão realizadas. A melhoria no tempo de execução pode ser vista na Tabela 3.1, mostrada abaixo.

Tabela 3.1 Tempo de detecção médio (ms)

k	Tons de cinza	Borda	Total
400	11,90	13,51	25,48
300	9,09	10,72	19,79
200	6,23	7,82	14,04
100	2,92	4,51	7,55
50	1,22	2,95	4,12
25	0,62	2,24	3,10

3.2 Classificador baseado em reconstrução modular

A idéia de detecção de pedestres baseado em classificadores mais simples foi proposta por Papageorgiou C., et al. [Moh01]. No artigo, que tinha como objetivo construir um classificador de pedestres baseado em componentes, Papageorgiou criou detectores para quatro regiões diferentes na imagem de um pedestre: cabeça e ombros, pernas, braço direito e braço esquerdo. A saída dos quatro classificadores era então combinada utilizando SVM (*Support Vector Machine*) para a construção do classificador de pedestres.

O classificador proposto neste trabalho visa unir as idéias de classificação baseada em reconstrução proposta Malagón-Borja e Fuentes, com uma adaptação da idéia de PCA Modular, que visa extrair informações de partes separadas do corpo humano.

3.2.1 Utilização de PCA Modular

A utilização de PCA Modular para a detecção baseada em reconstrução segue a idéia original, onde supõe-se que se obtermos matrizes de projeção de determinadas regiões de um pedestre, poderemos ter uma reconstrução total melhor. A qualidade superior desta reconstrução deve-se ao fato de que os PCs são extraídos de regiões menores, e assim estão menos sujeitos a problemas como a oclusão parcial, já que apenas algumas regiões sofreriam com a oclusão, enquanto outras não sofreriam alteração.

A técnica proposta divide a imagem de um pedestre em três partes: cabeça, torso e pernas. Para cada uma três sub-imagens obtemos uma matriz de reconstrução que representa a estrutura média de uma cabeça humana, a estrutura média de um torso humano e a estrutura média das pernas humanas. Desta maneira cada classificador pode se especializar em uma determinada região, ao invés de um classificador que tente modelar a estrutura de um pedestre. Então, a saída de cada um dos três classificadores é combinada para formar a saída do classificador de pedestres.

A divisão das imagens de treinamento, cujas dimensões originais eram 45×105 , é feita levando em consideração a média das imagens presentes no conjunto de treinamento. Através de análise foi constatado que a cabeça fica melhor representada numa sub-imagem de dimensões 45×20 extraída da parte superior da imagem, o torso fica

melhor representado na sub-imagem de dimensões 45×40 extraída logo abaixo da cabeça e as pernas bem representadas com o restante da imagem, com dimensões 45×45 . A divisão de uma imagem e suas respectivas imagens de borda podem ser vistas na Figura 3.4.



Figura 3.4 Divisão de uma imagem de pedestre para a classificação baseada em PCA Modular. Da esquerda para a direita: a imagem original, suas sub-imagens, imagem de borda e suas sub-imagens.

O comportamento esperado das imagens de não-pedestres é análogo ao do classificador proposto por Malagón-Borja e Fuentes, ou seja, sua utilidade será em termos um erro para comparar com o erro de reconstrução das imagens de pedestre. Apesar de não fazer sentido o uso dos termos *cabeça*, *torso* e *pernas* para não-pedestres, eles serão utilizados no resto do documento para facilitar o entendimento do classificador, com a distinção de mostrá-los em itálico>.

3.2.2 Treinamento

A fase de treinamento do classificador proposto cria, a partir dos quatro conjuntos utilizados por Malagón-Borja e Fuentes, doze conjuntos: *cabeça*, *torso* e *pernas* de pedestres em tons de cinza; *cabeça*, *torso* e *pernas* de imagens de borda de pedestres; *cabeça*, *torso* e *pernas* de não-pedestres em tons de cinza; e *cabeça*, *torso* e *pernas* de imagens de borda de não-pedestres.

O PCA é então aplicado para cada um desses doze conjuntos obtendo respectivamente as matrizes de projeção P_{gpC} , P_{gpT} , P_{gpP} , P_{gnC} , P_{gnT} , P_{gnP} , P_{epC} , P_{epT} , P_{epP} , P_{enC} , P_{enT} , P_{enP} ; e as médias μ_{gpC} , μ_{gpT} , μ_{gpP} , μ_{gnC} , μ_{gnT} , μ_{gnP} , μ_{epC} , μ_{epT} , μ_{epP} , μ_{enC} , μ_{enT} , μ_{enP} .

Os sub-índices das matrizes de projeção e das médias tem os seguintes significados, além dos explicados na Seção 3.1.1: *C* significa que a matriz e a média se referem a sub-imagens de cabeça; *T*, que se referem a sub-imagens de torso e *P*, que se referem a sub-imagens de pernas.

3.2.3 O classificador

Na fase de classificação, dada uma imagem em tons de cinza g realizamos o seguinte procedimento:

1. Dividimos a imagem em três sub-imagens g_C, g_T, g_P
2. Obtemos as imagens de borda de cada sub-imagem, obtendo e_C, e_T, e_P
3. Realizamos as doze reconstruções:

- $r_{gpC} = P_{gpC}P_{gpC}^T(g_C - \mu_{gpC}) + \mu_{gpC}$
- $r_{gpT} = P_{gpT}P_{gpT}^T(g_T - \mu_{gpT}) + \mu_{gpT}$
- $r_{gpP} = P_{gpP}P_{gpP}^T(g_P - \mu_{gpP}) + \mu_{gpP}$
- $r_{epC} = P_{epC}P_{epC}^T(e_C - \mu_{epC}) + \mu_{epC}$
- $r_{epT} = P_{epT}P_{epT}^T(e_T - \mu_{epT}) + \mu_{epT}$
- $r_{epP} = P_{epP}P_{epP}^T(e_P - \mu_{epP}) + \mu_{epP}$
- $r_{gnC} = P_{gnC}P_{gnC}^T(g_C - \mu_{gnC}) + \mu_{gnC}$
- $r_{gnT} = P_{gnT}P_{gnT}^T(g_T - \mu_{gnT}) + \mu_{gnT}$
- $r_{gnP} = P_{gnP}P_{gnP}^T(g_P - \mu_{gnP}) + \mu_{gnP}$
- $r_{enC} = P_{enC}P_{enC}^T(e_C - \mu_{enC}) + \mu_{enC}$
- $r_{enT} = P_{enT}P_{enT}^T(e_T - \mu_{enT}) + \mu_{enT}$
- $r_{enP} = P_{enP}P_{enP}^T(e_P - \mu_{enP}) + \mu_{enP}$

4. Calculamos os errors de reconstrução:

- $d_{gpC} = |r_{gpC} - g_C|$
- $d_{gpT} = |r_{gpT} - g_T|$
- $d_{gpP} = |r_{gpP} - g_P|$
- $d_{epC} = |r_{epC} - e_C|$
- $d_{epT} = |r_{epT} - e_T|$
- $d_{epP} = |r_{epP} - e_P|$
- $d_{gnC} = |r_{gnC} - g_C|$
- $d_{gnT} = |r_{gnT} - g_T|$
- $d_{gnP} = |r_{gnP} - g_P|$
- $d_{enC} = |r_{enC} - e_C|$

- $d_{enT} = |r_{enT} - e_T|$
- $d_{enP} = |r_{enP} - e_P|$

5. Calculamos os erros totais para cada sub-imagem:

- $d_{tC} = d_{gnC} + d_{enC} - (d_{gpC} + d_{epC})$
- $d_{tT} = d_{gnT} + d_{enT} - (d_{gpT} + d_{epT})$
- $d_{tP} = d_{gnP} + d_{enP} - (d_{gpP} + d_{epP})$

6. Calculamos o erro total da imagem:

$$d_t = d_{tC} + d_{tT} + d_{tP} \quad (3.3)$$

7. Aplicamos a seguinte regra para classificar g :

$$classe(g) = \begin{cases} \text{pedestre} & \text{se } d_t \geq 0 \\ \text{n\~{a}o-pedestre} & \text{se } d_t < 0 \end{cases}$$

3.2.4 Discuss\~{a}o

O estudo da t\~{e}cnica de PCA Modular mostra que a reconstru\~{c}o\~{a}o utilizando-a \u00e9 melhor do que a reconstru\~{c}o\~{a}o baseada em PCA. Este ind\u00edcio nos leva a crer que um classificador baseado em reconstru\~{c}o\~{a}o que utilize o PCA Modular obtenha resultados melhores do que um classificador que utilize apenas o PCA.

O c\u00e1lculo do PCA nas sub-imagens pode apresentar como vantagem, al\u00e9m da melhor reconstru\~{c}o\~{a}o, a possibilidade da cria\~{c}o\~{a}o de um classificador que utilize apenas alguns dos classificadores locais. Al\u00e9m do classificador que utiliza as tr\u00eas sub-imagens, este trabalho tamb\u00e9m avaliar\u00e1 os classificadores que utilizam: apenas a cabe\~{c}a e o torso; apenas o torso e as pernas; e apenas a cabe\~{c}a e as pernas. Tais classificadores s\u00e3o descritos no Ap\u00eêndice A.

Como visto na Se\~{c}o\~{a}o 3.1.3, a introdu\~{c}o\~{a}o de diferentes pesos na equa\~{c}o\~{a}o de c\u00e1lculo do erro total 3.1, \u00e9 justificada pelo entendimento de que nem todos os erros tem a mesma relev\u00e2ncia na constitui\~{c}o\~{a}o do erro total. Analogamente, podemos dar import\u00e2ncias diferentes a sa\u00edda de cada um dos classificadores locais, utilizando a equa\~{c}o\~{a}o 3.4.

$$d_t = w_{tC}d_{tC} + w_{tT}d_{tT} + w_{tP}d_{tP} \quad (3.4)$$

Al\u00e9m das investiga\~{c}o\~{e}s citadas, tamb\u00e9m \u00e9 interessante verificar se a redund\u00e2ncia de informa\~{c}o\~{e}s das imagens em tons de cinza com as imagens de borda vista em [CMRC10] tamb\u00e9m \u00e9 observada no classificador baseado em reconstru\~{c}o\~{a}o modular.

CAPÍTULO 4

Experimentos

A metodologia utilizada para a avaliação do classificador proposto é de suma importância e será explicada, assim como os resultados obtidos pelo classificador, neste capítulo.

A Seção 4.1 explica os passos tomados para a criação da base de dados utilizada e como foi feita a divisão das bases em vários conjuntos de treinamento e teste; a Seção 4.2 descreve os parâmetros utilizados para o Algoritmo Genético e sua função de aptidão; a Seção 4.3 descreve uma métrica através da qual será possível comparar os diferentes classificadores; por fim, a Seção 4.4 descreve os resultados obtidos pelas diferentes versões do classificador proposto e verifica a validade da hipóteses levantadas na Subseção 3.2.4.

4.1 Base de dados

4.1.1 Imagens de pedestres

Os experimentos foram executados sobre a base MIT CBCL Pedestrian Database [cbc00]. A base contém 924 imagens coloridas de pedestres em posições frontal e traseira, de tamanho 64×128 . Para este estudo todas as imagens da base foram recortadas para o tamanho de 45×105 , que é aproximadamente o menor tamanho em que todos os pedestres são representados sem cortes.

Após o ajuste no tamanho as imagens foram convertidas para escala de cinza utilizando a função `rgb2gray` do MATLAB. Esta função retira da imagem as informações de tom e saturação, mantendo a informação de luminosidade (Y), através da equação 4.1, onde R é a intensidade de vermelho; G a intensidade de verde e B a intensidade de azul para cada pixel da imagem.

$$Y = 0,2989 * R + 0,5870 * G + 0,1140 * B \quad (4.1)$$

A equação 4.1 leva em conta a função de luminosidade, que descreve a sensibilidade média de um ser humano para diferentes comprimentos de onda. Essa função foi padronizada pela *Commission internationale de l'éclairage* (CIE) e sua análise indica que o sistema visual humano é mais sensível a perceber luminância da cor verde, depois da vermelha e por último da azul, justificando os pesos utilizados no cálculo [WS00].

4.1.2 Imagens de não pedestres

Para a construção da base negativa, ou seja, de não pedestres, foram utilizadas cento e vinte imagens de diversos tamanho que não continham nenhum pedestre. Estas imagens eram subdivididas em pedaços de dimensões 45×105 . Ao fim deste processo, cinco mil imagens foram escolhidas aleatoriamente para a construção do conjunto negativo.

A diferença no tamanho dos conjuntos é inerente ao problema, pois o conjunto negativo deve representar todos os objetos que não sejam pedestres. Esta não é uma particularidade de detecção de pedestres, mas é decorrente do fato de que o problema de detecção de objetos é naturalmente assimétrico.

4.1.3 Conjuntos de treinamento e teste

As bases foram divididas na proporção de 75% para treinamento e 25% para testes. Para validar a técnica utilizada, esta divisão foi feita aleatoriamente e repetida três vezes, obtendo três grupos, cada um com 693 imagens de pedestres e 3750 imagens de não pedestres para treinamento e 231 imagens de pedestres e 1250 de não pedestres para teste.

4.2 Utilização do Algoritmo Genético

Algoritmos Genéticos foram utilizados para estimar os pesos da equação de cálculo de erro total 3.4. A implementação utilizada foi a presente no MATLAB, que objetiva minimizar a função que é fornecida como parâmetro. Os valores dos outros parâmetros do Algoritmo Genético, como número máximo de gerações, probabilidade de *crossover* e de mutação foram os definidos como padrão pelo MATLAB.

De acordo com a formulação do MATLAB a função de aptidão de cada cromossomo deve ser minimizada. A função utilizada classifica toda a base de *treinamento* usando o procedimento descrito na Seção 3.2.3; as imagens mal classificadas são contabilizadas, e a quantidade total (T) representa a aptidão do indivíduo.

É importante ressaltar que apesar da formulação inicial de Algoritmos Genéticos ser baseada no conceito da sobrevivência do mais apto, a formulação do MATLAB minimiza a aptidão dos indivíduos. Caso seja utilizada uma implementação que vise maximizar a aptidão, bastaria calcular a aptidão final do indivíduo como $1/T$, ao invés de T .

4.3 Avaliação dos Classificadores

Para que seja possível a comparação de classificadores diferentes, é necessária a utilização de uma métrica comum. A utilizada neste trabalho é área sob a curva ROC (AUC, do inglês *Area Under the ROC Curve*) [Faw06]. Essa métrica foi escolhida por

ser amplamente utilizada na área de reconhecimento de padrões. Antes da explicação da curva ROC é fundamental entender os conceitos de verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo.

Considere um problema de classificação binária, ou seja, exemplos são classificados como pertencentes a uma ou outra classe. Sem perda de generalidade, diremos que os exemplos podem ser classificados como *positivos* ou *negativos*. Neste caso, para a classificação de um exemplo são possíveis quatro resultados:

- O exemplo é positivo e a classificação o indica como positivo, chamado verdadeiro positivo (TP, do inglês *True Positives*);
- O exemplo é negativo e a classificação o indica como positivo, chamado falso positivo (FP, do inglês *False Positives*);
- O exemplo é negativo e a classificação o indica como negativo, chamado verdadeiro negativo (TN, do inglês *True Negatives*);
- O exemplo é positivo e a classificação o indica como negativo, chamado falso negativo (FN, do inglês *False Negatives*).

A curva ROC (*Receiver Operating Characteristic*) é um gráfico utilizado para a visualização da taxa de verdadeiros positivos pela taxa de falsos positivos num sistema de classificação binária enquanto o limiar de discriminação é alterado. A taxa de verdadeiros positivos (TPR), também conhecida como sensibilidade, é calculada por $TPR = TP / (TP + FN)$ e representa a porcentagem de exemplos positivos reconhecidos pelo classificador. A taxa de falsos positivos (FPR), também conhecida como especificidade, é calculada pela equação $FPR = FP / (FP + TN)$ e representa a porcentagem de exemplos negativos reconhecidos pelo classificador.

A AUC é uma métrica utilizada para facilitar a comparação de curvas ROC. Seu significado pode ser visto como uma nota dada ao classificador. Como os valores das taxas de verdadeiro positivo e falso positivo estão entre 0 e 1, a curva ROC pode ter área máxima 1, ou seja, um classificador com $AUC = 1$ é considerado ótimo [Faw06]. Apesar de ser possível que um classificador com AUC alta tenha uma performance pior em uma determinada região do que outro com AUC menor, a AUC ainda assim é uma boa medida para a avaliação de um classificador.

4.4 Análise das hipóteses

Esta Seção demonstrará os experimentos realizados para comprovar as quatro hipóteses levantadas na Seção 3.2.4, e discutirá os resultados obtidos, analisando-os a fim de provar a validade de cada hipótese.

As quatro hipóteses são: um classificador baseado em PCA modular obtém resultados melhores que um classificador baseado apenas em PCA; classificadores que levam em consideração apenas algumas sub-regiões da imagem podem ser construídos;

a introdução de pesos na equação 3.3, para diferenciar a contribuição de cada sub-imagem, traz resultados melhores; e, imagens em tons de cinza e imagens de borda tem redundância também nessa abordagem.

Para verificar a validade das hipóteses, os classificadores foram treinados para cada conjunto e as imagens de testes de cada conjunto foram classificadas. Além da utilização dos três conjuntos, as classificações foram realizadas utilizando os seguintes valores para o parâmetro k : 25, 50, 100, 150 e 200.

4.4.1 Equivalência dos classificadores

De modo a validar a hipótese de que um classificador baseado em PCA modular é melhor que um baseado em PCA, analisaremos a equivalência das quatro variações do classificador encontradas em [CMRC10]: o classificador original; o classificador que utiliza apenas imagens em tons de cinza; o classificador que utiliza apenas imagens de borda; e o classificador com pesos.

A análise da área sob a curva ROC (AUC) do classificador original mostrada na Figura 4.1 mostra que o classificador que baseia-se na reconstrução do PCA Modular tem uma diminuição maior do valor da AUC de acordo com o aumento do k .

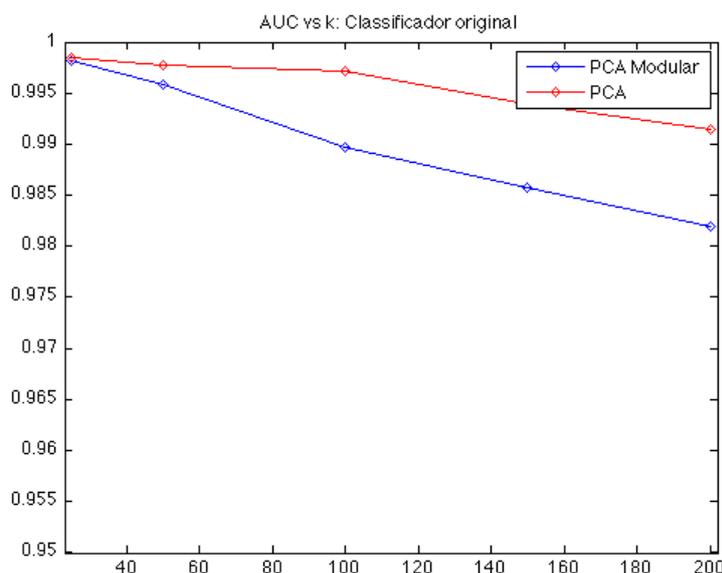


Figura 4.1 Gráfico da AUC vs k para o classificador original. Apesar da diminuição no valor, a AUC do classificador baseado em PCA modular ainda é próxima do classificador baseado em PCA. É importante ressaltar a escala do gráfico e o fato de que a diferença entre as AUCs é no máximo de 0.01.

O comportamento observado para o classificador original também é observado para os outros classificadores. A diferença é maior principalmente nos classificadores

Tabela 4.1 AUC vs k para classificadores com imagens em tons de cinza

k	PCA Modular	PCA
25	0,9947	0,9956
50	0,9903	0,9952
100	0,9791	0,9913
150	0,9735	0,9852
200	0,9673	0,9828

Tabela 4.2 AUC vs k para classificadores com pesos

k	PCA Modular	PCA
25	0,9977	0,9987
50	0,9953	0,9973
100	0,9884	0,9905
150	0,9841	0,9938
200	0,9791	0,9926

que levam em consideração apenas imagens em tons de cinza e nos classificadores com peso. O único classificador que pode ser visto como exceção é o baseado em imagens de borda, que apesar de ainda apresentar a diminuição vista nos outros classificadores, ainda consegue manter-se estável. Os valores das AUCs podem ser vistos nas tabelas 4.1, 4.2 e 4.3, respectivamente.

Uma explicação para o decréscimo do valor da AUC com o aumento do valor de k se deve à quantidade de autovetores na matriz de covariância e sua relação com k . Para o PCA normal, uma imagem de 45×105 gera uma matriz de covariância de 4.725×4.725 , que resulta em 4725 autovetores. O maior valor de k testado (200) representa apenas 4,23% dos autovetores. Para o PCA Modular cada sub-imagem é representada por matriz de dimensões 45×20 , 45×40 e 45×45 , que resultam, respectivamente, em

Tabela 4.3 AUC vs k para classificadores com imagens de borda

k	PCA Modular	PCA
25	0,9955	0,9966
50	0,9942	0,9968
100	0,9945	0,9967
150	0,9936	0,9954
200	0,9925	0,9953

900, 1800 e 2025 autovetores. O mesmo valor citado anteriormente para k representa 22,22%, 11,11% e 9,8% dos autvetores.

Esses autovalores indicam que mais PCs estão sendo incluídos no classificador baseado em PCA Modular, o que pode introduzir ruído ao sistema. O ruído vem do fato de estarmos incluindo autovetores com um relativo baixo autovalor associado, como explicado na Seção 2.3.1, ou seja, estaremos incluindo novos eixos que pouco colaboram com o aumento da variância.

4.4.2 Redundância das imagens tons de cinza e de borda

A redundância das imagens em tons de cinza com as imagens de borda analisada em [CMRC10] também é verificada para a reconstrução baseada em PCA Modular. Como pode ser visto na Figura 4.2, apesar de ter a AUC maior para valores menores de k , o classificador que utiliza os dois tipos de imagens tem uma queda mais acentuada do que o classificador de utiliza apenas imagens de borda.

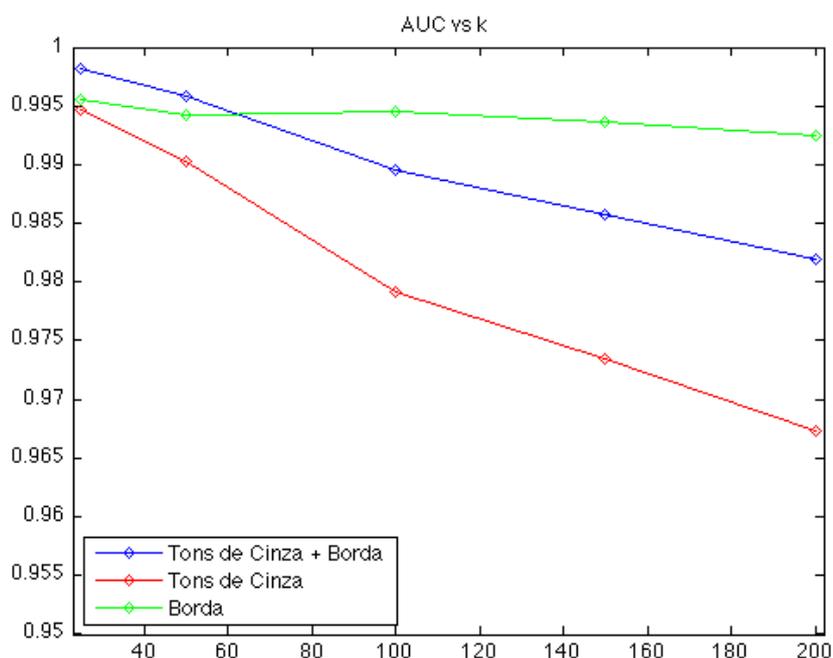


Figura 4.2 O gráfico mostra a variação da AUC de acordo com o k para a classificação baseada em imagens em tons de cinza e de borda, apenas para imagens em tons de cinza e apenas para imagens de borda.

4.4.3 Classificadores de partes

Para avaliar a utilidade de classificadores que baseiam-se apenas em algumas das sub-imagens, devemos compará-los ao classificador que utiliza as três sub-imagens, e compará-los entre si. Essa comparação pode ser vista na Figura 4.3. Os classificadores parciais foram construídos como descrito no Apêndice A e levam em consideração: a cabeça e as pernas; a cabeça e o torso; e, o torso e as pernas. Os classificadores serão chamados de cabeça/pernas, cabeça/torso e torso/pernas, respectivamente, de modo a facilitar o entendimento.

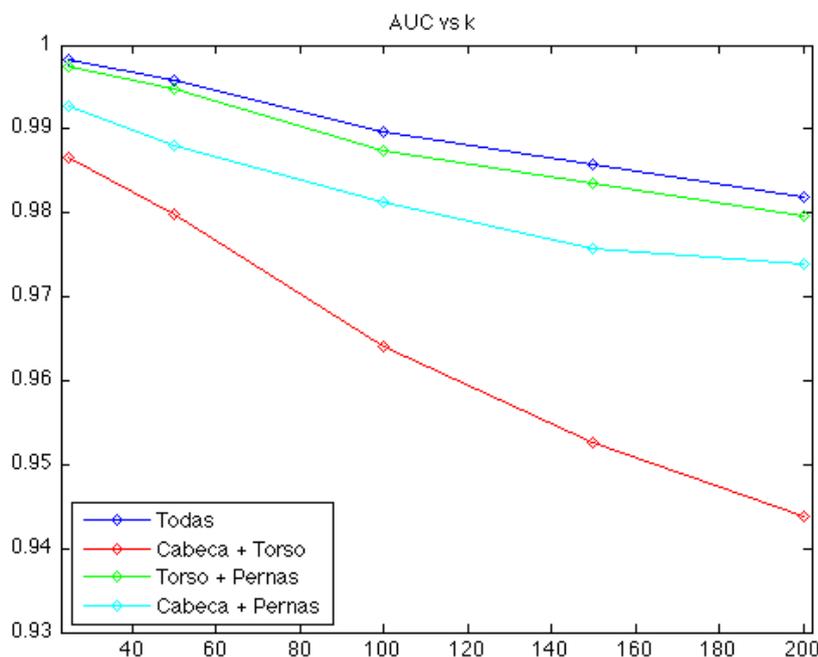


Figura 4.3 O gráfico mostra a variação da AUC de acordo com o k para os classificadores baseados em sub-imagens.

A análise da Figura 4.3 mostra que os apenas os classificadores baseadoS nas sub-imagens torso/pernas e cabeça/pernas obtêm resultados comparáveis ao do classificador que utiliza as três sub-imagens. O classificador cabeça/torso apresenta um bom AUC para $k = 25$, porém o valor cai acentuadamente com o aumento do k .

Uma possível explicação para esse resultado pode ser encontrada analisando o trabalho de Papageorgiou [Moh01]. Ao examinar os detectores distintos propostos por ele, o que obteve o melhor resultado individual foi o das pernas, seguido pelos detectores dos braços e por fim o detector da cabeça. Analogamente, os dois classificadores que utilizam a sub-imagem que continha as pernas do pedestre obtiveram resultados melhores e o classificador que não utiliza a sub-imagem das pernas teve a pior performance. Isso mostra que o torso e as pernas bastam para modelar um pedestre.

Tabela 4.4 Verdadeiros positivos (TP) para classificadores total e torso/pernas

<i>VP</i>	Todas	Torso/Pernas
25	0,9423	0,9293
50	0,8802	0,8989
100	0,8037	0,8297
150	0,7648	0,8066
200	0,7532	0,7922

Tabela 4.5 Tempo de detecção médio para reconstrução modular (ms)

<i>k</i>	Todas	Torso/pernas
200	13,10	12,33
100	7,84	5,87
50	5,26	4,12
25	4,04	3,26

Apesar de obter uma AUC menor do que o classificador que utiliza as três sub-imagens, ao analisarmos outros dados, como a quantidade de verdadeiros positivos obtidos pelo classificador torso/pernas, vemos que ele se mantém mais estável do que o classificador completo. Os valores dos verdadeiros positivos para os dois classificadores pode ser visto na Tabela 4.4.

A análise dos verdadeiros positivos demonstra que ao aumentarmos o valor de k , o classificador que utiliza todas as sub-imagens tende a classificar erradamente mais pedestres, enquanto o classificador torso/pernas também apresenta essa tendência, porém menos acentuada. Além disso, o classificador torso/pernas, por realizar menos reconstruções, consegue ser mais rápido que o classificador que utiliza todas as sub-imagens, como poder visto na Tabela 4.5.

4.4.4 Classificação modular com pesos

Como visto em [CMRC10] e na Seção 3.1.3 a introdução de pesos na equação do erro total 3.1 resultou em uma melhor classificação. Nesta seção investigaremos se a utilização de pesos para discriminar a contribuição de cada sub-imagem na reconstrução total obterá melhorias similares.

Novamente recorreremos às curvas ROC para analisar o comportamento do classificador com pesos e sem pesos. A Figura 4.4 mostra as curvas ROC para um dos conjuntos, com $k = 50$.

Experimentos também foram executados para avaliar a utilização de pesos para os classificadores parciais, porém seus resultados foram piorados com a adição dos pesos. A Tabela 4.6 mostra as AUCs dos classificadores parciais com e sem pesos para

os valores de k testados. Para nenhum valor de k os classificadores parciais com peso tiveram desempenho melhor ou igual ao dos que não utilizavam pesos.

Tabela 4.6 AUCs para classificadores parciais sem e com a utilização pesos

K	Torso/pernas		Cabeça/pernas		Cabeça/torso	
	Sem pesos	Com pesos	Sem pesos	Com pesos	Sem pesos	Com pesos
25	0,9974	0,9967	0,9928	0,9916	0,9866	0,9845
50	0,9948	0,9946	0,9881	0,9862	0,9799	0,9768
100	0,9874	0,9867	0,9812	0,9782	0,9641	0,9623
150	0,9836	0,9818	0,9757	0,9712	0,9527	0,9505
200	0,9796	0,9773	0,9739	0,9688	0,9435	0,9387

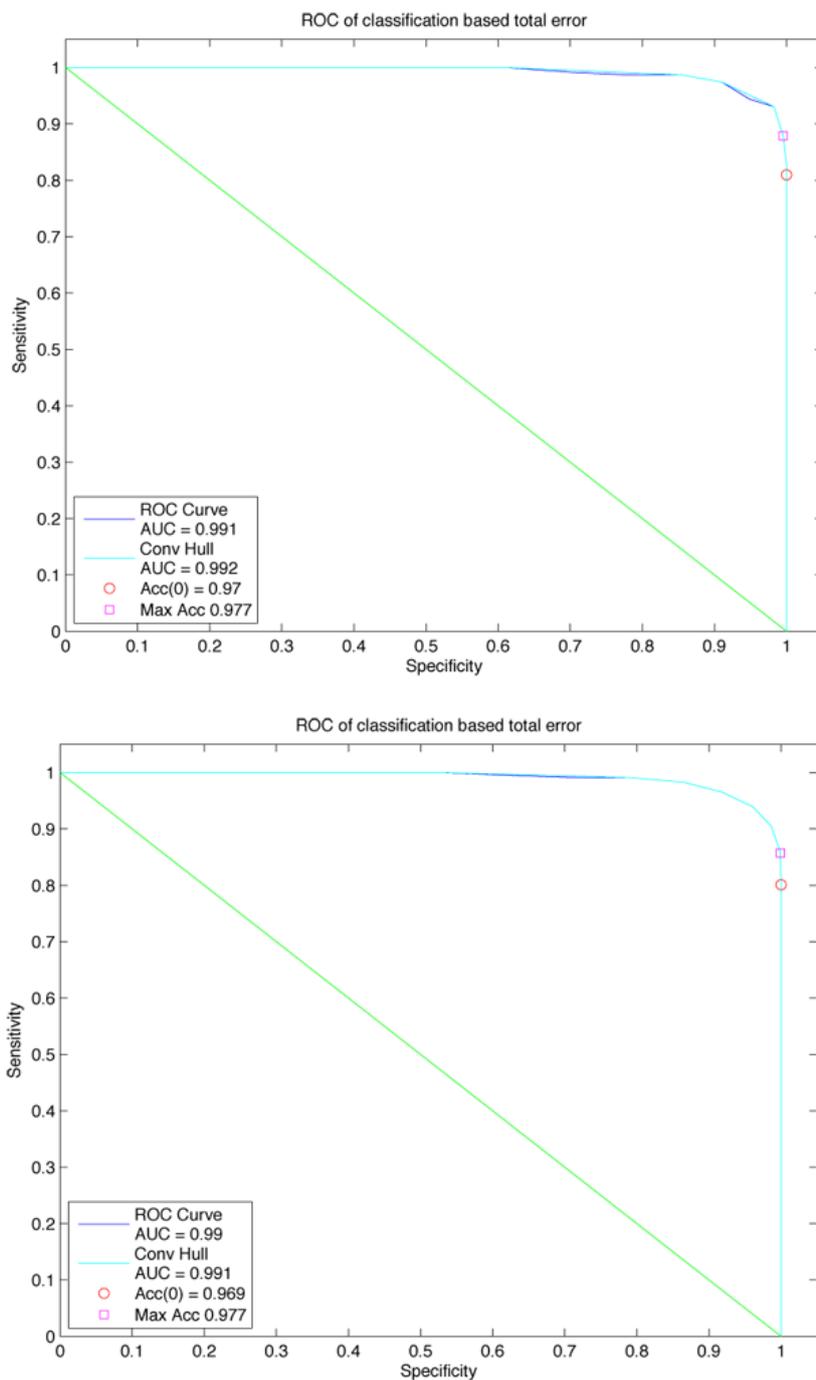


Figura 4.4 A figura de cima mostra a curva ROC do classificador baseado em PCA modular sem pesos; A figura de baixo mostra a curva ROC do classificador com pesos. As duas curvas foram calculadas $k = 50$. Para os outros valores de k as curvas apresentaram o mesmo padrão: o classificador sem pesos tem resultados melhores do que o com pesos.

Conclusão

A proposta de um classificador de pedestres que baseia-se no erro de reconstrução criou um novo uso para a técnica de PCA. A idéia da utilização de PCA para calcular a reconstrução foi analisada e melhorada em [CMRC10], com a proposta da utilização de pesos para diferenciar a contribuição de cada erro de reconstrução. Também foi proposta a não utilização de imagens em tons de cinza, já que as imagens de borda continham informações suficientes para uma boa classificação.

Baseado nessas idéias, e na de PCA Modular, este trabalho propôs a criação de um classificador que utiliza os erros de reconstrução de componentes de um pedestre. A proposta parte do pressuposto de que reconstruções realizadas de maneira modular obtêm melhores resultados.

Os resultados obtidos no Capítulo 4 mostraram que, apesar de não obtermos resultados melhores, obtemos resultados equivalentes. Além disso comprovamos que a redundância entre as imagens em tons de cinza e as imagens de borda são mantidas também para esta técnica, o que possibilita a criação de um classificador mais rápido.

A maior contribuição, porém, deste trabalho é a construção de um classificador que utiliza apenas informações do torso e das pernas de um pedestre, e que apesar de não utilizar toda a informação contida na imagem, tem uma taxa de classificação bastante próxima das taxas de um classificador que utiliza a totalidade da imagem. A grande vantagem de utilizar apenas o torso e as pernas é o ganho em desempenho, já que, na área de detecção de pedestres, o desempenho é um requisito fundamental.

Para trabalhos futuros, pretendemos analisar a performance do classificador ao utilizarmos outras técnicas para a reconstrução, como o 2D-PCA [YZFyY04]. Além disso, pretendemos construir um sistema completo de detecção de pedestres, que possa ser empregado para a detecção de pedestres em vídeos.

Formulação dos classificadores parciais

Este apêndice descreve os classificadores baseados em apenas algumas da sub-imagens, citados na seção 3.2.4.

A.1 Classificador cabeça/torso

A.1.1 Treinamento

A partir dos quatro conjuntos utilizados por Malagón-Borja e Fuentes são criados oito conjuntos: cabeça e torso de pedestres em tons de cinza; cabeça e torso de imagens de borda de pedestres; *cabeça* e *torso* de não-pedestres em tons de cinza; e *cabeça* e *torso* de imagens de borda de não-pedestres.

O PCA é então aplicado para cada um desses oito conjuntos obtendo respectivamente as matrizes de projeção $P_{gpC}, P_{gpT}, P_{gnC}, P_{gnT}, P_{epC}, P_{epT}, P_{enC}, P_{enT}$; e as médias $\mu_{gpC}, \mu_{gpT}, \mu_{gnC}, \mu_{gnT}, \mu_{epC}, \mu_{epT}, \mu_{enC}, \mu_{enT}$.

A.1.2 O classificador

Na fase de classificação, dada uma imagem em tons de cinza g realizamos o seguinte procedimento:

1. Dividimos a imagem em duas sub-imagens g_C, g_T
2. Obtemos as imagens de borda de cada sub-imagem, obtendo e_C, e_T
3. Realizamos as oito reconstruções:

- $r_{gpC} = P_{gpC}P_{gpC}^T(g_C - \mu_{gpC}) + \mu_{gpC}$
- $r_{gpT} = P_{gpT}P_{gpT}^T(g_T - \mu_{gpT}) + \mu_{gpT}$
- $r_{epC} = P_{epC}P_{epC}^T(e_C - \mu_{epC}) + \mu_{epC}$
- $r_{epT} = P_{epT}P_{epT}^T(e_T - \mu_{epT}) + \mu_{epT}$
- $r_{gnC} = P_{gnC}P_{gnC}^T(g_C - \mu_{gnC}) + \mu_{gnC}$
- $r_{gnT} = P_{gnT}P_{gnT}^T(g_T - \mu_{gnT}) + \mu_{gnT}$
- $r_{enC} = P_{enC}P_{enC}^T(e_C - \mu_{enC}) + \mu_{enC}$

- $r_{enT} = P_{enT}P_{enT}^T(e_T - \mu_{enT}) + \mu_{enT}$

4. Calculamos os errors de reconstrução:

- $d_{gpC} = |r_{gpC} - g_C|$
- $d_{gpT} = |r_{gpT} - g_T|$
- $d_{epC} = |r_{epC} - e_C|$
- $d_{epT} = |r_{epT} - e_T|$
- $d_{gnC} = |r_{gnC} - g_C|$
- $d_{gnT} = |r_{gnT} - g_T|$
- $d_{enC} = |r_{enC} - e_C|$
- $d_{enT} = |r_{enT} - e_T|$

5. Calculamos os erros totais para cada sub-imagem:

- $d_{tC} = d_{gnC} + d_{enC} - (d_{gpC} + d_{epC})$
- $d_{tT} = d_{gnT} + d_{enT} - (d_{gpT} + d_{epT})$

6. Calculamos o erro total da imagem:

$$d_t = d_{tC} + d_{tT} \quad (\text{A.1})$$

7. Aplicamos a seguinte regra para classificar g :

$$classe(g) = \begin{cases} \text{pedestre} & \text{se } d_t \geq 0 \\ \text{n\~{a}o-pedestre} & \text{se } d_t < 0 \end{cases}$$

Para a utilizaç~{a}o de pesos, ajustamos a equaç~{a}o [A.1](#), para:

$$d_t = w_{tC}d_{tC} + w_{tT}d_{tT} \quad (\text{A.2})$$

A.2 Classificador cabeça/pernas

A.2.1 Treinamento

A partir dos quatro conjuntos utilizados por Malagón-Borja e Fuentes são criados oito conjuntos: cabeça e pernas de pedestres em tons de cinza; cabeça e pernas de imagens de borda de pedestres; *cabeça* e *pernas* de não-pedestres em tons de cinza; e *cabeça* e *pernas* de imagens de borda de não-pedestres.

O PCA é então aplicado para cada um desses oito conjuntos obtendo respectivamente as matrizes de projeção P_{gpC} , P_{gpP} , P_{gnC} , P_{gnP} , P_{epC} , P_{epP} , P_{enC} , P_{enP} ; e as médias μ_{gpC} , μ_{gpP} , μ_{gnC} , μ_{gnP} , μ_{epC} , μ_{epP} , μ_{enC} , μ_{enP} .

A.2.2 O classificador

Na fase de classificação, dada uma imagem em tons de cinza g realizamos o seguinte procedimento:

1. Dividimos a imagem em duas sub-imagens g_C, g_P
2. Obtemos as imagens de borda de cada sub-imagem, obtendo e_C, e_P
3. Realizamos as oito reconstruções:

- $r_{gpC} = P_{gpC}P_{gpC}^T(g_C - \mu_{gpC}) + \mu_{gpC}$
- $r_{gpP} = P_{gpP}P_{gpP}^T(g_P - \mu_{gpP}) + \mu_{gpP}$
- $r_{epC} = P_{epC}P_{epC}^T(e_C - \mu_{epC}) + \mu_{epC}$
- $r_{epP} = P_{epP}P_{epP}^T(e_P - \mu_{epP}) + \mu_{epP}$
- $r_{gnC} = P_{gnC}P_{gnC}^T(g_C - \mu_{gnC}) + \mu_{gnC}$
- $r_{gnP} = P_{gnP}P_{gnP}^T(g_P - \mu_{gnP}) + \mu_{gnP}$
- $r_{enC} = P_{enC}P_{enC}^T(e_C - \mu_{enC}) + \mu_{enC}$
- $r_{enP} = P_{enP}P_{enP}^T(e_P - \mu_{enP}) + \mu_{enP}$

4. Calculamos os errors de reconstrução:

- $d_{gpC} = |r_{gpC} - g_C|$
- $d_{gpP} = |r_{gpP} - g_P|$
- $d_{epC} = |r_{epC} - e_C|$
- $d_{epP} = |r_{epP} - e_P|$
- $d_{gnC} = |r_{gnC} - g_C|$
- $d_{gnP} = |r_{gnP} - g_P|$
- $d_{enC} = |r_{enC} - e_C|$
- $d_{enP} = |r_{enP} - e_P|$

5. Calculamos os erros totais para cada sub-imagem:

- $d_{tC} = d_{gnC} + d_{enC} - (d_{gpC} + d_{epC})$
- $d_{tP} = d_{gnP} + d_{enP} - (d_{gpP} + d_{epP})$

6. Calculamos o erro total da imagem:

$$d_t = d_{tC} + d_{tP} \quad (\text{A.3})$$

7. Aplicamos a seguinte regra para classificar g :

$$classe(g) = \begin{cases} \text{pedestre} & \text{se } d_t \geq 0 \\ \text{n\~{a}o-pedestre} & \text{se } d_t < 0 \end{cases}$$

Para a utilizaç~{a}o de pesos, ajustamos a equa~{c}o~{a}o [A.3](#), para:

$$d_t = w_{tC}d_{tC} + w_{tP}d_{tP} \quad (\text{A.4})$$

A.3 Classificador torso/pernas

A.3.1 Treinamento

A partir dos quatro conjuntos utilizados por Malagón-Borja e Fuentes s~{a}o criados oito conjuntos: torso e pernas de pedestres em tons de cinza; torso e pernas de imagens de borda de pedestres; *torso* e *pernas* de n~{a}o-pedestres em tons de cinza; e *torso* e *pernas* de imagens de borda de n~{a}o-pedestres.

O PCA é ent~{a}o aplicado para cada um desses oito conjuntos obtendo respectivamente as matrizes de projeç~{a}o $P_{gpT}, P_{gpP}, P_{gnT}, P_{gnP}, P_{epT}, P_{epP}, P_{enT}, P_{enP}$; e as m~{e}dias $\mu_{gpT}, \mu_{gpP}, \mu_{gnT}, \mu_{gnP}, \mu_{epT}, \mu_{epP}, \mu_{enT}, \mu_{enP}$.

A.3.2 O classificador

Na fase de classifica~{c}o~{a}o, dada uma imagem em tons de cinza g realizamos o seguinte procedimento:

1. Dividimos a imagem em duas sub-imagens g_T, g_P
2. Obtemos as imagens de borda de cada sub-imagem, obtendo e_T, e_P
3. Realizamos as oito reconstru~{c}o~{e}s:

- $r_{gpT} = P_{gpT}P_{gpT}^T(g_T - \mu_{gpT}) + \mu_{gpT}$
- $r_{gpP} = P_{gpP}P_{gpP}^T(g_P - \mu_{gpP}) + \mu_{gpP}$
- $r_{epT} = P_{epT}P_{epT}^T(e_T - \mu_{epT}) + \mu_{epT}$
- $r_{epP} = P_{epP}P_{epP}^T(e_P - \mu_{epP}) + \mu_{epP}$
- $r_{gnT} = P_{gnT}P_{gnT}^T(g_T - \mu_{gnT}) + \mu_{gnT}$
- $r_{gnP} = P_{gnP}P_{gnP}^T(g_P - \mu_{gnP}) + \mu_{gnP}$
- $r_{enT} = P_{enT}P_{enT}^T(e_T - \mu_{enT}) + \mu_{enT}$
- $r_{enP} = P_{enP}P_{enP}^T(e_P - \mu_{enP}) + \mu_{enP}$

4. Calculamos os errors de reconstrução:

- $d_{gpT} = |r_{gpT} - g_T|$
- $d_{gpP} = |r_{gpP} - g_P|$
- $d_{epT} = |r_{epT} - e_T|$
- $d_{epP} = |r_{epP} - e_P|$
- $d_{gnT} = |r_{gnT} - g_T|$
- $d_{gnP} = |r_{gnP} - g_P|$
- $d_{enT} = |r_{enT} - e_T|$
- $d_{enP} = |r_{enP} - e_P|$

5. Calculamos os erros totais para cada sub-imagem:

- $d_{tT} = d_{gnT} + d_{enT} - (d_{gpT} + d_{epT})$
- $d_{tP} = d_{gnP} + d_{enP} - (d_{gpP} + d_{epP})$

6. Calculamos o erro total da imagem:

$$d_t = d_{tT} + d_{tP} \quad (\text{A.5})$$

7. Aplicamos a seguinte regra para classificar g :

$$classe(g) = \begin{cases} \text{pedestre} & \text{se } d_t \geq 0 \\ \text{n\~{a}o-pedestre} & \text{se } d_t < 0 \end{cases}$$

Para a utilizaç~{a}o de pesos, ajustamos a equa~{c}o~{a}o [A.5](#), para:

$$d_t = w_{tT}d_{tT} + w_{tP}d_{tP} \quad (\text{A.6})$$

Referências Bibliográficas

- [Bol86] José Luiz. et al. Boldrini. *Álgebra Linear*. Harper & Row do Brasil, 1986. 2.1
- [cbc00] Mit cbcl pedestrian database, 2000. 4.1.1
- [CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967. 2.4.1
- [CMRC10] Guilherme V. Carvalho, Lailson B. Moraes, Tsang I. R., and George D. C. Cavalcanti. A weighted image reconstruction based on pca for pedestrian detection. *Workshops of SIBGRAPI*, 2010. 3, 3.1.3, 3.1.3, 3.2.4, 4.4.1, 4.4.2, 4.4.4, 5
- [Cut00] L.S. Cutler, R.; Davis. Robust real-time periodic motion detection, analysis, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2000. 1
- [DTS06] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, volume 3952 of *Lecture Notes in Computer Science*, pages 428–441. Springer Berlin / Heidelberg, 2006. 1
- [FA10] A. Frank and A. Asuncion. UCI Machine Learning Repository, 2010. 2.6.4
- [Faw06] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ROC Analysis in Pattern Recognition. 4.3
- [GA04] Rajkiran Gottumukkal and Vijayan K. Asari. An improved face recognition technique based on modular pca approach. *Pattern Recognition Letters*, 25(4):429 – 436, 2004. 1, 2.4
- [Ger09] A.M.; Sappa A.D.; Graf T. Gerónimo, D.; López. Survey of pedestrian detection for advanced driver assistance systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2009. 1
- [GWE03] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003. 2.5

- [Hol92] John H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992. 2.6
- [Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417 – 441, 1933. 2.3
- [Kan07] Feiqi Kang, Wenxiong; Deng. Research on intelligent visual surveillance for public security. *Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on*, 2007. 1
- [LCY⁺09] Yazhou Liu, Xilin Chen, Hongxun Yao, Xinyi Cui, Chaoran Liu, and Wen Gao. Contour-motion feature (cmf): A space-time approach for robust pedestrian detection. *Pattern Recognition Letters*, 30(2):148 – 156, 2009. 1
- [Lee02] W.E.L. Lee, L.; Grimson. Gait analysis for recognition and classification. *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, 2002. 1
- [Mar88] Minc H. Marcus, M. *Introduction to Linear Algebra*. Dover, 1988. 2.1
- [MBF09] Luis Malagón-Borja and Olac Fuentes. Object detection using image reconstruction with pca. *Image and Vision Computing*, 27(1-2):2 – 9, 2009. Canadian Robotic Vision 2005 and 2006. 1, 3, 3.1
- [Mit96] Melanie Mitchell. *Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1996. 2.6.2
- [Moh01] C.; Poggio T. Mohan, A.; Papageorgiou. Example-based object detection in images by components. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2001. 1, 3.2, 4.4.3
- [MSZ04] Krystian Mikolajczyk, Cordelia Schmid, and Andrew Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In Tomás Pajdla and Jirí Matas, editors, *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 69–82. Springer Berlin / Heidelberg, 2004. 1
- [Pea01] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572, 1901. 2.3
- [Sha04] Y.; Hayun G. Shashua, A.; Gdalyahu. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. *Intelligent Vehicles Symposium, 2004 IEEE*, 2004. 1
- [Shl09] J. Shlens. A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego*, 2009. 2.3, 2.3

- [SK87] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, Mar 1987. 2.3.2
- [TK09] S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Elsevier/Academic Press, 2009.
- [TP91] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–89, 1991. 2.3.2, 3.1
- [VJS05] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63:153–161, 2005. 10.1007/s11263-005-6644-8. 1
- [WS00] Günther Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley-Interscience, 2000. 4.1.1
- [YZFyY04] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing yu Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:131–137, 2004. 5
- [ZT98] Djemel Ziou and Salvatore Tabbone. Edge detection techniques - an overview. *International Journal of Pattern Recognition and Image Analysis*, 8:537–559, 1998. 2.5