



Universidade Federal de Pernambuco
Centro de Informática

Pós-graduação em Ciência da Computação

Detecção de Pedestres com Oclusão

Silvio Gustavo de Oliveira Santos

Dissertação de Mestrado

Recife

Agosto de 2013

Universidade Federal de Pernambuco
Centro de Informática

Silvio Gustavo de Oliveira Santos

Detecção de Pedestres com Oclusão

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Centro de Informática da Uni-
versidade Federal de Pernambuco como requisito parcial
para obtenção do grau de Mestre em Ciência da Computa-
ção.*

Orientador: *Prof. Tsang Ing Ren*

Co-orientador: *Prof. George Darmiton da Cunha Cavalcanti*

Recife

Agosto de 2013

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Santos, Silvio Gustavo de Oliveira
Detecção de pedestres com oclusão / Silvio Gustavo de
Oliveira Santos. - Recife: O Autor, 2013.
xii, 50 f.: il., fig., tab.

Orientador: Tsang Ing Ren.
Dissertação (mestrado) - Universidade Federal de Pernambuco.
CIn, Ciência da Computação, 2013.

Inclui referências.

1. Processamento de imagem. 2. Visão computacional. I. Ren,
Tsang Ing (orientador). II. Título.

621.367

CDD (23. ed.)

MEI2013 – 143

Dissertação de Mestrado apresentada por **Silvio Gustavo de Oliveira Santos** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **“Detecção de Pedestres com Oclusão”** orientada pelo Prof. George Darmiton da Cunha Cavalcanti e aprovada pela Banca Examinadora formada pelos professores:

Profa. Veronica Teichrieb
Centro de Informática / UFPE

Prof. Wellington Pinheiro dos Santos
Departamento de Engenharia Biomédica/ UFPE

Prof. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 19 de agosto de 2013

Profa. Edna Natividade da Silva Barros
Coordenadora da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Agradecimentos

Aos meus pais, pois sempre acreditaram na minha capacidade e nunca mediram esforços para me dar uma boa educação, a fim de que eu adquirisse os valores corretos que um ser humano deve ter.

A Deus, por sempre se mostrar presente em minha vida, me carregando nos momentos difíceis e me guiando nos momentos em que a dúvida e a incerteza estavam presentes.

Ao professor Tsang, pelo apoio e orientação durante o mestrado.

*A mente que se abre a uma nova ideia
jamais voltará ao seu tamanho original.*

—ALBERT EINSTEIN

Resumo

A detecção de pedestres é uma área muito promissora no ramo da visão computacional, pois possibilita aplicações importantes em sistemas de auxílio à direção de veículos, sistemas de vigilância e na área da robótica. Muitas técnicas novas surgiram com grandes melhorias nas taxas de detecção, mas ainda assim, seus desempenhos caem consideravelmente quando os pedestres estão sofrendo oclusão. Este trabalho analisa como características baseadas no HOG são construídas e como elas são afetadas quando os pedestres estão parcialmente ocluídos. Como o HOG é aplicado em muitos sistemas de detecção de pedestres, entender como a oclusão afeta a sua performance é importante para prever como extratores de características derivados dele se comportarão na mesma situação. Mais especificamente, o HOG, HOG-LBP e duas novas combinações de características, HOG-LTP e HOG-LMEBP foram analisados. Para tal, oclusões foram geradas sinteticamente mesclando tamanhos diferentes de imagens de não pedestres, em diferentes direções, com a base de dados INRIA. O resultado foi uma nova base de dados composta por oclusões nas direções ascendente, descendente e lateral (da esquerda para direita). Os quatro extratores de características foram usados para treinar classificadores SVM na base de dados INRIA e foram testados com a nova base de dados com oclusão. Os resultados confirmam que o desempenho dos classificadores decai estritamente à medida que a oclusão aumenta, porém o impacto depende de que área do pedestre está escondida. Nós também mostramos que é possível melhorar as taxas de detecção em condições de oclusão simplesmente combinando classificadores.

Palavras-chave: pedestre, detecção, oclusão, combinação, visão computacional.

Abstract

Pedestrian detection is still a very promising area in computer vision, since it enables interesting applications for car assistance, vigilance systems and robots. Lots of new techniques appeared with great improvements on detection rates, but their performance completely deteriorates when pedestrians are under occlusion. This work analyses how HOG based feature extractors are constructed and how they are affected when pedestrians are partially occluded. Since HOG is applied in a lot of pedestrian detection systems, understanding how occlusion affects its performance is useful to predict how its derived feature extractors behave in same situation. More specifically, HOG, HOG-LBP and two new feature combinations, HOG-LTP and HOG-LMEBP, were analyzed. To accomplish this task, occlusions were synthetically generated merging different sizes of non-pedestrian images, in different directions, with the INRIA dataset. The result was a new dataset composed by occlusions in bottom-up, top-down and left-right directions. The four feature extractors were used to train SVM classifiers on INRIA dataset and they were tested on the new occluded dataset. The results confirm that detection rates strictly decreases when occlusion area grows, but its impact depends on which pedestrian area is hidden. We also show that we can improve the detection rates under occlusion situations simply combining classifiers.

Keywords: pedestrian, detection, occlusion, combination, computer vision.

Sumário

1	Introdução	1
1.1	Detecção de pedestres	2
1.2	Motivação	4
1.3	Objetivos	4
1.4	Estrutura da dissertação	5
2	Detecção de Pedestres	6
2.1	Aquisição e representação da imagem digital	8
2.2	Pré-processamento	9
2.3	Extratores de características	10
2.3.1	HOG	10
2.3.2	HOG-LBP	15
2.3.3	LTP	17
2.3.4	LMEBP	17
2.4	Classificação	20
3	Método Proposto	23
3.1	Combinação de características	23
3.2	Combinação de classificadores	24
3.2.1	Voto majoritário	25
3.2.2	Classificador em cascata	25
3.3	Base com oclusão sintética	26
4	Experimentos	29
4.1	Experimentos na base original usando combinação de características	29

4.2	Experimentos na base degradada usando combinação de características	33
4.2.1	Oclusão ascendente	33
4.2.2	Oclusão descendente	34
4.2.3	Oclusão lateral	35
4.2.4	Oclusão misturada	37
4.3	Experimentos na base degradada usando combinação de classificadores	38
4.3.1	Oclusão ascendente	38
4.3.2	Oclusão descendente	39
4.3.3	Oclusão lateral	41
4.3.4	Oclusão misturada	41
4.3.5	Imagens de não pedestres	43
5	Conclusões e Trabalhos Futuros	45

Lista de Figuras

1.1	Exemplo de imagens de pedestres da base INRIA [1].	3
1.2	Fluxo geral de detecção de pedestres.	3
2.1	Processo de aquisição da imagem. Adaptado de [2].	8
2.2	Processo de convolução para imagens digitais. Um filtro é aplicado para toda a imagem, multiplicando seus coeficientes $w(x,y)$ pelas intensidades dos <i>pixels</i> de cada região da imagem. O resultado é somado e substituído no elemento central da região da imagem onde o filtro foi aplicado. Adaptado de [2].	12
2.3	Divisão da imagem em células, indicadas pelos quadrados pretos, e blocos, formados pelos quadrados vermelhos.	14
2.4	Extração do LBP de uma região 3x3 da imagem. As intensidades dos <i>pixels</i> vizinhos são subtraídas do <i>pixel</i> central e a limiarização é aplicada. Se a diferença é maior ou igual a zero, é atribuído o valor 1, caso contrario, atribuí-se 0. Adaptado de [3].	16
2.5	Imagens de pedestres e suas respectivas imagens LBP.	16
2.6	Extração do LTP de uma região 3x3 para um limiar igual a 5.	18
2.7	Exemplo do cálculo do LMEBP em uma região 3x3. Na esquerda, o <i>pixel</i> da região central com valor 6 e seus vizinhos. À direita, a diferença entre os vizinhos e o <i>pixel</i> central. Adaptado de [4].	19
2.8	Bordas ordenadas em função da sua magnitude e a codificação binária correspondente, atribuindo 0 para bordas negativas e 1 para bordas positivas [4].	20
2.9	Resultado do cálculo do LMEBP [4].	20

2.10	A figura (b) mostra a imagem LBP extraída da figura (a) e as figuras de (c) à (f) mostram as imagens obtidas a partir da 1ª maior borda até a 4ª maior borda do LMEBP [4].	21
2.11	Exemplo da separação de duas classes utilizando SVM. A reta r encontrada pelo SVM separa as classes com margem máxima, enquanto a reta tracejada s separa com uma margem pequena.	22
3.1	Separação do LTP em códigos LBP superior e inferior [5].	24
3.2	Combinação de classificadores utilizando voto majoritário.	25
3.3	Combinação de classificadores utilizando classificador em cascata.	26
3.4	Imagens da base criada com oclusão sintética. A imagem (a) possui oclusão ascendente, enquanto a (b) tem oclusão descendente e a (c) oclusão lateral.	27
3.5	A primeira imagem original da base INRIA e exemplos de imagens de pedestres dos 9 conjuntos formados pelas oclusões ascendentes.	28
4.1	Curva ROC obtida para o HOG quando treinado e testado com imagens sem oclusão.	31
4.2	Curva ROC obtida para o HOG-LBP quando treinado e testado com imagens sem oclusão.	31
4.3	Curva ROC obtida para o HOG-LTP quando treinado e testado com imagens sem oclusão.	32
4.4	Curva ROC obtida para o HOG-LMEBP quando treinado e testado com imagens sem oclusão.	32
4.5	Comparativo das 4 técnicas quando submetidas à oclusão ascendente.	34
4.6	Comparativo das 4 técnicas quando submetidas à oclusão descendente.	36
4.7	Comparativo das 4 técnicas quando submetidas à oclusão lateral.	37
4.8	Comparativo entre as 3 técnicas isoladas e combinadas quando submetidas à oclusão ascendente.	40
4.9	Comparativo entre as 3 técnicas isoladas e combinadas quando submetidas à oclusão descendente.	41

4.10 Comparativo entre as 3 técnicas isoladas e combinadas quando submetidas à oclusão lateral.	42
---	----

Lista de Tabelas

4.1	Resultados dos experimentos executados na base de dados original, sem oclusão.	30
4.2	Resultados dos experimentos executados na base de dados formada pelos conjuntos de oclusão ascendente.	33
4.3	Resultados dos experimentos executados na base de dados formada pelos conjuntos de oclusão descendente.	35
4.4	Resultados dos experimentos executados na base de dados formada pelos conjuntos de oclusão lateral.	36
4.5	Resultados dos experimentos executados nas bases de dados formadas por imagens sem oclusão misturadas com imagens com oclusões de diferentes tamanhos.	38
4.6	Resultados dos experimentos executados na base com oclusões ascendentes utilizando combinação de classificadores.	39
4.7	Resultados dos experimentos executados na base com oclusões descendentes utilizando combinação de classificadores.	40
4.8	Resultados dos experimentos executados na base com oclusões laterais utilizando combinação de classificadores.	42
4.9	Resultados dos experimentos executados sobre o conjunto misturado utilizando combinação de classificadores.	43
4.10	Resultados dos experimentos executados com os classificadores isolados sobre o conjunto de imagens de não pedestres.	43
4.11	Resultados dos experimentos executados com os classificadores combinados sobre o conjunto de imagens de não pedestres.	44

CAPÍTULO 1

Introdução

A quantidade de imagens geradas por câmeras de vigilância nunca foi tão grande. Por questões de segurança, hoje, muitas residências e grande parte dos estabelecimentos comerciais são filmados 24 horas por dia. Porém, as imagens geradas por estes sistemas são subutilizadas, visto que normalmente são armazenadas e analisadas somente quando necessário ou precisam de fiscalização em tempo real. Com a evolução das técnicas de processamento de imagens, vem sendo possível extrair, em tempo real, informações cada vez mais relevantes das imagens. Deste modo, as imagens capturadas por câmeras de vigilância são capazes de prover uma grande quantidade de informação sobre o ambiente em que estão localizadas. Hoje em dia, já é possível encontrar sistemas que identificam o fluxo de pessoas em grandes áreas, provendo informações que auxiliam no gerenciamento do ambiente. Ainda na área de vigilância, é possível criar sistemas capazes de interpretar o comportamento das pessoas e identificar atitudes suspeitas. Tais sistemas usam como base técnicas de detecção e rastreamento de pedestres.

A detecção de pedestres é uma tarefa muito importante no campo da visão computacional, visto que essa área faz parte do futuro de tecnologias empregadas na vigilância, automação veicular e robótica [6]. Além das aplicações em vigilância citadas, em veículos ela é usada como sistema de segurança a fim de evitar colisões com pedestres que eventualmente apareçam na frente ou atrás do automóvel [7] e em robótica é utilizada como parte de sistemas responsáveis na interação entre robôs e pessoas.

Apesar das técnicas de detecção já terem sido bastante aprimoradas [8], ainda há muito a ser feito, porque mesmo os sistemas com melhor taxa de acerto possuem seu desempenho completamente deteriorado quando imagens ruidosas, de baixa resolução e com oclusão são analisadas. Os dois primeiros problemas são mais fáceis de resolver, visto que são mais controláveis e podem ser minimizados com câmeras melhores e através de técnicas de pré-processamento, porém a oclusão é um problema mais difícil de tratar. Em algumas situações, o pedestre pode

estar ocluído por carros, por placas, árvores ou até mesmo por outros pedestres. Muito esforço tem sido feito para lidar com situações de oclusão, mas ainda há muito a ser feito, já que é difícil prever quando um pedestre está sendo ocluído, qual parte dele não está visível e o que o está ocluindo.

1.1 Detecção de pedestres

A detecção de pedestres consiste em procurar em uma imagem indivíduos que estão em pé, podendo estar de frente, de costas ou até mesmo de lado para a câmera [8]. A abordagem mais utilizada é a de varrer exaustivamente a imagem utilizando janelas de tamanhos diferentes, visto que os indivíduos podem aparecer de diversos tamanhos, dependendo da proximidade que estão da câmera. Cada região da imagem sob a janela de busca é submetida a um classificador que indica se a região corresponde ou não a um pedestre.

O processo de desenvolvimento de um detector de pedestres é constituído em geral por duas fases. A primeira é a de treinamento, quando um classificador é treinado para distinguir imagens de pedestres de imagens de não pedestres. A segunda fase é a de testes, onde o desempenho do classificador é avaliado através de um conjunto de imagens diferente das utilizadas durante o treinamento. A Figura 1.1 mostra exemplos de imagens de pedestres da base INRIA [1].

Durante a fase de treinamento, um conjunto somente com imagens de pedestres e outro com imagens de não pedestres é usado para treinar o classificador como distinguir dentre esses dois grupos. Antes de ser enviada ao classificador, cada imagem normalmente passa por um processo chamado pré-processamento, onde é tratada para ajustar o seu contraste, cor e reduzir ruídos causados pela má iluminação, ou para ressaltar características que serão utilizadas na fase seguinte. Em seguida, as imagens são submetidas a um extrator de características, que é responsável por extrair informações da imagem que auxiliarão na identificação de um pedestre. Essas características podem ser, por exemplo, de textura, cor, forma, bordas ou até mesmo uma combinação dessas e de outras. A última etapa consiste em usar as características extraídas



Figura 1.1: Exemplo de imagens de pedestres da base INRIA [1].

para treinar o classificador. Para cada conjunto de características de uma imagem é atribuído um rótulo para informar ao classificador a que classe ela pertence, isto é, se corresponde a um pedestre ou não. O classificador então utiliza essas informações para aprender a diferenciar as características de pedestres das de não pedestres.



Figura 1.2: Fluxo geral de detecção de pedestres.

Após o treinamento, é possível avaliar o desempenho do classificador verificando como ele se comporta quando são apresentadas imagens diferentes daquelas usadas na fase de treinamento. A Figura 1.2 exhibe os passos executados para identificar pedestres em uma imagem.

1.2 Motivação

A detecção de pedestres em imagens é de grande interesse para o comércio e empresas de segurança, por exemplo. Sistemas de detecção podem ser usados como base para realizar a contagem e, conseqüentemente, levantar estatísticas de ocupação e de hábitos de consumidores em estabelecimentos comerciais. Em sistemas de segurança, a detecção de pedestres pode ser capaz de verificar se há pessoas ocupando uma área de acesso restrito, se há mais pessoas em um ambiente do que o permitido ou ser usado como base para sistemas que interpretam o comportamento humano, afim de identificar situações de perigo. Sistemas de auxílio à direção de veículos também se beneficiam da detecção de pedestres. Em sistemas como esses, câmeras no veículo detectam a presença de pedestres e são capazes de dar alertas ou modificar a velocidade do veículo para evitar acidentes. Estes são apenas alguns exemplos reais que motivam o estudo da detecção de pedestres.

Além dos problemas reais, a motivação para o estudo da detecção de pedestres se dá também pelo seu desafio. Apesar do grande avanço nos estudos e na melhoria das técnicas de detecção, ainda há muitos problemas a serem resolvidos, principalmente os relacionados à oclusão. A oclusão de pedestre se dá quando outros objetos da cena se localizam entre o pedestre e a câmera, escondendo partes do seu corpo. Quando o corpo do pedestre está ocluído, características importantes são perdidas, e as suas características visíveis acabam se misturando às do objeto que o está ocluindo. Assim, ocorre uma alta degradação nos resultados do sistema. Como não é fácil prever quando vai ocorrer, a oclusão se torna um grande deteriorador de sistemas de detecção. Diante desses motivos, a detecção de pedestres, e em particular, a oclusão na detecção, é uma área que merece bastante estudo e que ainda é possível contribuir significativamente.

1.3 Objetivos

O principal objetivo desse trabalho é avaliar como o desempenho de classificadores treinados com diferentes características é afetado quando submetidos a imagens com oclusão. Para tal,

uma base de dados com oclusão de diferentes tamanhos e direções foi criada a partir da combinação de imagens de pedestres com imagens de não pedestres. Assim, é possível entender como as partes não visíveis do pedestre influenciam na detecção dos classificadores. Mais especificamente, esse trabalho se propôs a:

- Realizar um levantamento e estudo do estado da arte das técnicas relacionadas à detecção de pedestres em imagens;
- Implementar técnicas relevantes no processo de detecção de pedestres;
- Combinar técnicas de detecção de pedestres, a fim de melhorar o desempenho na detecção;
- Criar uma base de imagens de pedestres com oclusão sintética de vários tamanhos e em diferentes direções;
- Realizar experimentos para avaliar o desempenho dos classificadores quando submetidos a imagens com e sem oclusão;
- Analisar os resultados obtidos.

1.4 Estrutura da dissertação

Este trabalho está organizado da seguinte forma: o capítulo 2 trata do estado da arte e de como é realizada a detecção de pedestres em imagens digitais. No capítulo 3, é apresentado o método proposto, que consiste na combinação de características e de classificadores, além de apresentar a base de imagens com oclusão sintética. Já no capítulo 4, são apresentados e discutidos os resultados dos experimentos utilizando os métodos originais e propostos. Por fim, o capítulo 5 trata das conclusões e trabalhos futuros.

Detecção de Pedestres

As técnicas mais eficientes de detecção de pedestres se baseiam em abordagens de janela de busca, onde pedestres são procurados exaustivamente na imagem através de uma janela que pode variar de tamanho, dependendo do tamanho mínimo e máximo que um pedestre pode alcançar. Um extrator de características bastante conhecido para a detecção de pedestres é o *Histogram of Oriented Gradients* (HOG) [1]. Ele atua extraindo informações de orientação e magnitude dos vetores gradiente das bordas do pedestre. Essas informações são utilizadas para construir um histograma que computa como na imagem de um pedestre as magnitudes dos vetores gradiente se distribuem de acordo com a sua orientação. Esse histograma é usado para treinar uma Máquina de Vetores de Suporte (*Support Vector Machine* - SVM), que decidirá se a imagem avaliada corresponde a um pedestre ou não. Por ter implementação simples e ao mesmo tempo apresentar bons resultados, o HOG é utilizado como base para outras técnicas, como a que combina o HOG com *Linear Binary Patterns* (LBP) [9]. O LBP extrai informações de textura que são invariantes à rotação e que, quando combinadas com o HOG, melhoram consideravelmente o desempenho na detecção, formando um novo extrator denominado HOG-LBP [10, 11]. Baseado no LBP, o *Local Maximum Edge Binary Patterns* (LMEBP) [4] é construído através da magnitude da diferença entre um *pixel* e seus vizinhos, conseguindo melhores resultados que o LBP na recuperação de imagens de textura.

A Análise de Componentes Principais (*Principal Component Analysis* - PCA) é aplicada em [12] mostrando que é possível detectar pedestres utilizando reconstrução de imagens. O PCA consiste em projetar um conjunto de dados para um novo sistema de coordenadas, cujos vetores da base desse novo sistema são ordenados em função do poder de representatividade que possuem sobre os dados. Assim, é possível projetar um conjunto de dados sobre uma quantidade menor de componentes que possuem grande representatividade sobre os dados. A ideia central em [12] é que a imagem resultante obtida ao projetar uma imagem de pedestre

sobre um conjunto de componentes principais extraídas de imagens de pedestres se assemelhará muito mais com um pedestre do que com qualquer outro objeto.

A transformada de Fourier é usada em [13] para calcular uma classe de filtros chamada *Average of Synthetic Exact Filters* (ASEF). Na detecção de pedestres [14], esse filtro é calculado a partir da média da resposta desejada no domínio espacial. Modelos baseados em partes do corpo são utilizados em [15] e [16], o que facilita a detecção quando o pedestre está sofrendo oclusão. Pequenos segmentos de linha ou de curva invariantes às diferenças de roupa e extraídos da silhueta do pedestre, chamados *edgelets* [16], são usados para detectar pedestres parcialmente ocluídos baseados nas partes do seu corpo. Diferente dos *edgelets* que utilizam um conjunto fixo de padrões de borda, os *shapelets* [17] aprendem automaticamente os padrões de borda que melhor discriminam imagens de pedestres das de não pedestres. Nesse caso, uma variação do AdaBoost desenvolvida por Viola e Jones [18] é usada para aprender os *shapelets* e para criar o classificador final a partir deles. Já os *shape contexts* [19, 20] são descritores que usam uma amostra de pontos internos e externos do contorno de um objeto para calcular um histograma baseado nos vetores entre um ponto de referência e os demais pontos da silhueta. Assim, pontos correspondentes em formas similares terão *shape context* similares, possibilitando o casamento de características e, conseqüentemente, a detecção. Viola e Jones [18] usam imagens integrais e AdaBoost para selecionar automaticamente um conjunto de classificadores fracos para a detecção de objetos, resultando em uma abordagem robusta para a detecção de pedestres.

Outras abordagens são baseadas na combinação desses e outros extratores, como [8, 21] que combinam características baseadas nos filtros de Haar, *shapelets*, *shape context* e HOG. Dollar estende [18] calculando imagens integrais de canais obtidos a partir de, por exemplo, formatos de cores diferentes, bordas e histogramas de magnitudes de gradientes [22]. Em outro trabalho [23], Dollar mostra que é possível otimizar a detecção em múltiplas escalas aproximando as características extraídas de uma escala próxima. Informações de cor, borda e textura são utilizadas em [24] para extrair um conjunto de características formado por mais de 170.000 dimensões. A dimensionalidade desse conjunto é reduzida através da projeção em um subespaço de 20 dimensões utilizando o método dos mínimos quadrados parciais.

2.1 Aquisição e representação da imagem digital

A imagem digital é resultado de um processo de amostragem e quantização realizado a partir de uma imagem analógica ou através da captura direta das imagens usando um sensor foto sensível. A Figura 2.1 mostra como é o processo de captura de imagens digitais. A luz que incide sobre o objeto que se quer capturar é refletida para as lentes da câmera, que por sua vez contém uma grande quantidade de sensores capazes de converter em sinais de saída proporcionais a sua intensidade. Esses sinais são convertidos para o formato digital e formam a imagem.

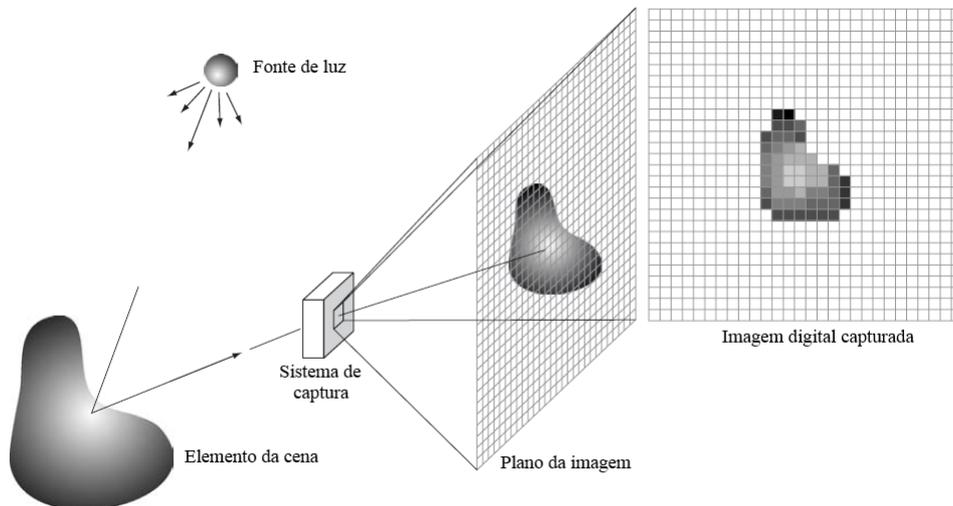


Figura 2.1: Processo de aquisição da imagem. Adaptado de [2].

A sua representação na forma digital consiste em uma função bidimensional de domínio discreto, normalmente representada por uma matriz, onde cada elemento contém informação de luminosidade da cena. Considerando $f(x, y)$ uma imagem digital representada por uma matriz de M linhas e N colunas, cada elemento da imagem pode ser determinado pelas suas coordenadas discretas x e y , onde $x < N$ e $y < M$. Além disso, geralmente adota-se a origem da imagem, o elemento $f(0, 0)$, como sendo o elemento do canto superior esquerdo da matriz.

Cada elemento da matriz corresponde a um ponto na imagem chamado de *pixel*. Como mencionado, cada *pixel* contém informações de luminosidade a respeito da imagem. No caso

de imagens em escala de cinza, cada *pixel* é composto de apenas uma componente que representa o nível de cinza daquele ponto na imagem. Em imagens coloridas, cada *pixel* é representado por mais de uma componente. Um dos modelos de cores mais usados é o RGB, onde cada *pixel* é composto por 3 componentes, representando as cores vermelho, verde e azul. O modelo de cor pode ser considerado como um espaço vetorial, onde os vetores de base são as suas componentes R, G e B. Assim, qualquer cor na imagem digital é representada como uma combinação linear das cores bases.

2.2 Pré-processamento

A primeira fase em um sistema de visão computacional, em geral, consiste em fazer um pré-processamento na imagem de forma a diminuir imperfeições e ressaltar características que serão usadas nas etapas seguintes.

A base de dados utilizada pelos classificadores expostos nesta seção é a INRIA [1]. Essa base é composta por imagens coloridas, com tamanhos de 96x160 *pixels* para as imagens de treinamento e de 70x134 *pixels* para as imagens de teste. Sendo assim, o primeiro passo na etapa de pré-processamento é o de normalização dos tamanhos das imagens da base, para que todas tenham dimensões de 64x128 *pixels*. Esse ajuste é realizado retirando-se 16 *pixels* de cada lado das imagens de treinamento e 3 *pixels* de cada lado das imagens de teste, a fim de normalizar o tamanho de todas as imagens e retirar regiões desnecessárias.

O segundo passo consiste em realizar uma transformação de cor, para que todas as imagens sejam convertidas para escalas de cinza. A conversão é realizada transformando as componentes RGB da imagem para o formato YUV e preservando somente a componente de luminância. A equação 2.1 [25] mostra que a luminância é obtida a partir de uma soma ponderada das componentes RGB, onde as variáveis *R*, *G* e *B* representam as componentes vermelho, verde e azul da imagem, respectivamente. Cada componente da imagem original é multiplicada por um peso específico e os resultados são somados, resultando na componente de luminância, que corresponde a imagem em escalas de cinza.

$$Y = 0.299R + 0.587G + 0.114B \quad (2.1)$$

2.3 Extratores de características

A função dos extratores de características é de capturar informações das imagens de pedestres que serão usadas nas etapas de treinamento e classificação. Essa fase é fundamental, pois a escolha das características influencia diretamente o resultado do sistema. As características variam de acordo com o que se quer identificar e podem ser de textura, forma, silhueta, cor, dentre outras. Para o caso de pedestres, as suas características mais importantes estão na sua silhueta, principalmente na região dos ombros e cabeça [17], pois são suas características mais marcantes e que melhor ajudam a diferenciar de outros objetos. Além da silhueta, características de textura também são comumente usadas e oferecem bons resultados.

Nesse trabalho, utilizamos extratores baseados no HOG. A razão dessa escolha, é que o HOG é usado como base para muitos extratores de sistemas de detecção de pedestres atuais, pois ele captura bem informações da silhueta do pedestre e apresenta bons resultados nos sistemas em que ele é aplicado [26]. Além do HOG, foi implementado uma variante chamada HOG-LBP, que além das características de silhueta, também captura informações de textura. Outros dois descritores de textura, o LTP e o LMEBP, também são apresentados nesse capítulo, pois suas combinações com o HOG serão abordadas no capítulo que tratamos do método proposto. Assim, quando avaliarmos como a oclusão afeta o HOG e sua combinação com esses demais extratores, poderemos ter uma ideia de como se comportam outros extratores baseados no HOG sob as mesmas condições.

2.3.1 HOG

O *Histogram of Oriented Gradients* (HOG) é um descritor que extrai as informações de orientação de bordas de uma imagem e as organiza em um vetor de histograma, onde cada posição

corresponde a um intervalo de ângulos e os valores correspondem à soma das magnitudes de cada vetor atribuído à determinada posição. Cada HOG é construído a partir das orientações de bordas contidas dentro de uma célula da imagem. As células, por sua vez, são organizadas em blocos que se sobrepõem. O descritor completo da imagem é formado a partir das informações contidas em todos os blocos da imagem.

A primeira etapa na construção do descritor consiste na conversão da imagem para escalas de cinza, realizada na etapa de pré-processamento. Em seguida, as componentes verticais e horizontais dos vetores gradiente da imagem são extraídas através do processo de convolução.

O processo de convolução em imagens bidimensionais consiste em aplicar operações sobre *pixels* em uma vizinhança determinada por uma janela que se desloca sobre a imagem. Para o caso de filtros lineares, os elementos da janela, chamados de pesos, são multiplicados pelos *pixels* correspondentes na imagem digital. Como exemplo, ao se aplicar um filtro linear com uma janela de tamanho 3x3 sobre um *pixel* localizado no ponto $f(x, y)$ de uma imagem f , obtém-se o seguinte resultado:

$$P = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1) \quad (2.2)$$

Onde P é o novo valor do *pixel* localizado no ponto (x, y) e $w(x, y)$ é a função que define os pesos da janela. A janela é centrada sobre o ponto (x, y) e o seu elemento central é o peso $w(0, 0)$. Em geral, a filtragem linear de uma imagem f de tamanho $M \times N$ por um filtro que possui uma janela de tamanho $m \times n$ é dado por:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x + s, y + t) \quad (2.3)$$

Onde $a = (m - 1)/2$ e $b = (n - 1)/2$. Deste modo, para aplicar o processo de filtragem a uma imagem representada por uma matriz de M linhas e N colunas, faz-se $x = 1, 2, 3, \dots, M$ e $y = 1, 2, 3, \dots, N$.

A Figura 2.2 ilustra o processo de convolução para uma imagem $f(x, y)$. Nesse caso, um filtro é aplicado a uma região de tamanho 3x3 da imagem. O filtro é representado por uma má-

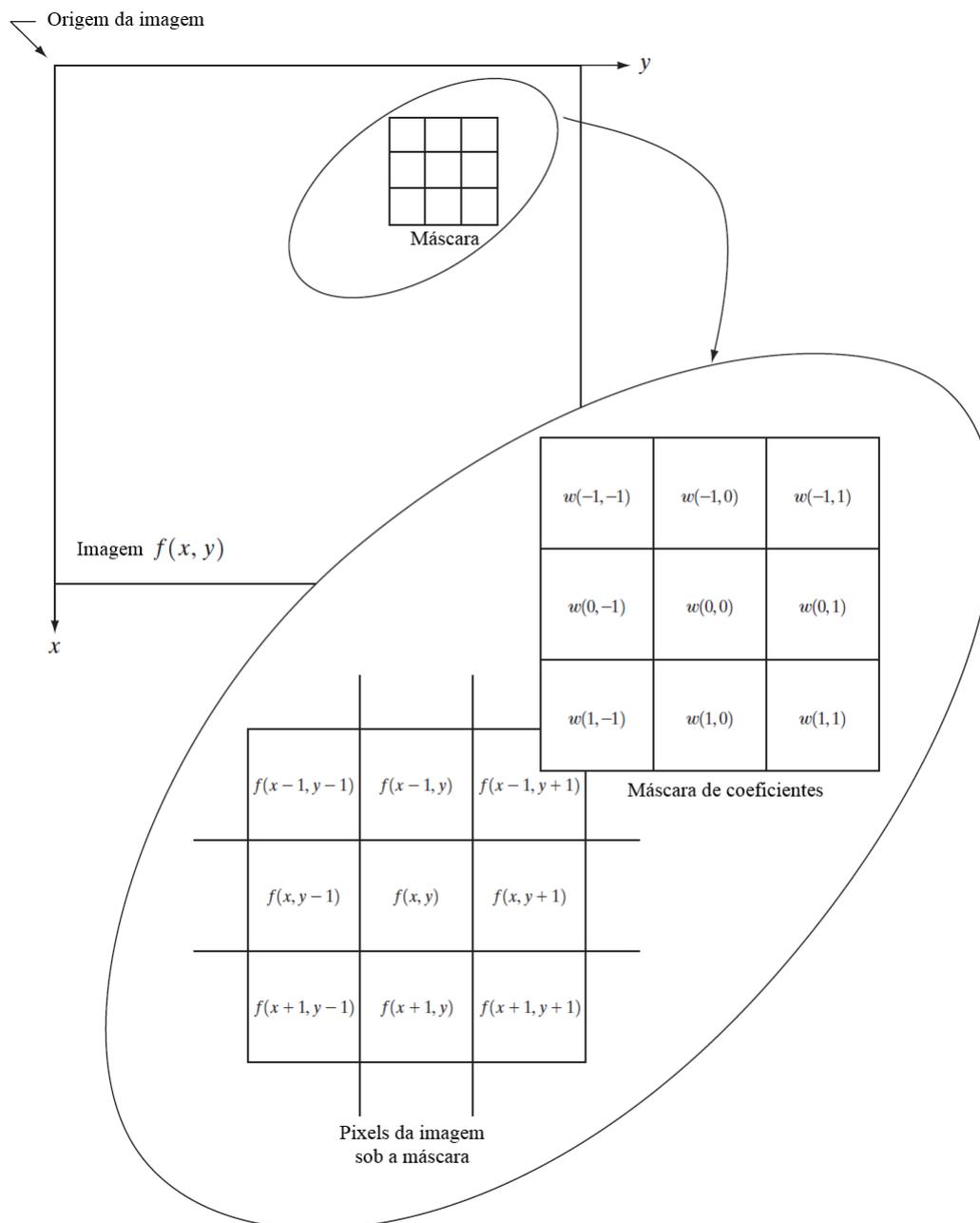


Figura 2.2: Processo de convolução para imagens digitais. Um filtro é aplicado para toda a imagem, multiplicando seus coeficientes $w(x, y)$ pelas intensidades dos *pixels* de cada região da imagem. O resultado é somado e substituído no elemento central da região da imagem onde o filtro foi aplicado. Adaptado de [2].

cara de coeficientes com pesos $w(x,y)$ que são multiplicados pelos elementos correspondentes da imagem e somados. O resultado é substituído no elemento central da região da imagem.

No caso do HOG, as máscaras dos filtros utilizadas para a extração das componentes dos vetores gradiente são mostradas nas expressões 2.4 e 2.5.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$G_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad (2.5)$$

As equações 2.6 e 2.7 mostram as componentes horizontal e vertical do vetor gradiente, respectivamente, sendo obtidas através da operação de convolução com os seus respectivos filtros.

$$I_h = I * G_x \quad (2.6)$$

$$I_v = I * G_y \quad (2.7)$$

A partir das componentes verticais e horizontais, obtemos a magnitude e a orientação do vetor gradiente a partir das equações demonstradas em 2.8 e 2.9, respectivamente.

$$M = \sqrt{I_h^2 + I_v^2} \quad (2.8)$$

$$\theta = \arctan \frac{I_v}{I_h} \quad (2.9)$$

A imagem é subdividida em blocos de 16×16 *pixels*, onde cada bloco contém células de 8×8 *pixels*. Os blocos se sobrepõem em uma célula, de modo que cada célula contribui para o cálculo do histograma de 4 blocos distintos. A Figura 2.3 mostra as regiões da imagem compreendidas pelas células e pelos seus respectivos blocos.

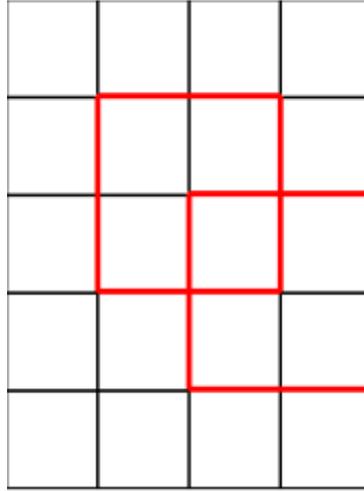


Figura 2.3: Divisão da imagem em células, indicadas pelos quadrados pretos, e blocos, formados pelos quadrados vermelhos.

Para cada célula de um bloco, um vetor de histograma é extraído, onde cada posição corresponde a uma faixa dos ângulos dos vetores gradiente e o valor corresponde à soma das magnitudes dos vetores com orientação correspondente. Dessa forma, o descritor funciona computando os votos de cada ângulo, ponderados pela sua magnitude. Os sinais dos ângulos calculados são desprezados, sendo computadas somente variações de 0° à 180° e o histograma é composto por 9 posições, onde cada uma equivale a uma variação de 20° [1].

O descritor completo da imagem é formado pela concatenação dos HOGs de cada bloco. Devido às diferenças de iluminação entre diferentes regiões na imagem, as células são normalizadas por bloco. A normalização é realizada utilizando L2-Hys [1], através da equação a seguir:

$$v_n = \frac{v}{\|v\|} \quad (2.10)$$

O vetor descritor normalizado tem os seus valores truncados para que não sejam maiores do que 0.2, de forma a diminuir a influência de regiões que possuam grande magnitude [27]. Após a truncagem, a norma é recalculada.

2.3.2 HOG-LBP

O HOG-LBP é um extrator de características formado pela combinação das informações de gradiente do HOG com as informações de textura obtidas através do *Linear Binary Patterns* (LBP) [3, 9, 28]. O LBP é um extrator de textura bastante usado, pois é invariante à rotação e a mudanças monotônicas do nível de cinza da imagem, além de ser computacionalmente eficiente, visto que seu cálculo é simples e rápido.

O LBP é calculado através da diferença de cada *pixel* de uma região 3x3 com o *pixel* central. A região de vizinhança pode ser definida de outros tamanhos [29], porém o HOG-LBP utiliza janelas com tamanho 3x3. Para cada diferença obtida, é aplicado um limiar de forma que para os resultados maiores ou iguais a zero, são atribuídos o valor 1 e para os demais, o valor 0, como definido na equação 2.11. Os valores resultantes são considerados um número binário e são convertidos para o formato decimal, resultando no LBP. A equação 2.12 mostra como o cálculo do LBP é realizado. A partir dos LBPs calculados para todas as regiões da imagem, é possível calcular um histograma que serve como descritor de textura.

A Figura 2.4 mostra um exemplo do cálculo do LBP para uma região 3x3 extraída da imagem. O *pixel* central é subtraído de cada vizinho e o limiar é aplicado resultando em um número binário que, quando convertido para decimal, representa o LBP da região.

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{caso contrário} \end{cases} \quad (2.11)$$

$$LBP = \sum_{i=1}^8 2^{(i-1)} f(I(g_i) - I(g_c)) \quad (2.12)$$

As imagens da Figura 2.5 mostram os resultados da aplicação do LBP à imagens de pedestres. É possível notar os padrões de textura nas regiões onde se encontram os pedestres que se diferenciam das demais regiões da imagem.

No HOG-LBP, o LBP é utilizado na forma de um histograma calculado a partir de padrões uniformes [29]. São considerados *Local Binary Patterns* uniformes aqueles que, quando contado de forma circular, possuem no máximo duas transições de 0 para 1 ou vice e versa. Como

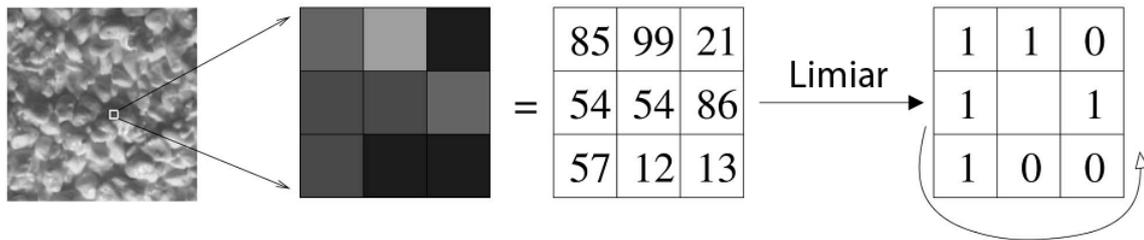


Figura 2.4: Extração do LBP de uma região 3x3 da imagem. As intensidades dos *pixels* vizinhos são subtraídas do *pixel* central e a limiarização é aplicada. Se a diferença é maior ou igual a zero, é atribuído o valor 1, caso contrário, atribuí-se 0. Adaptado de [3].

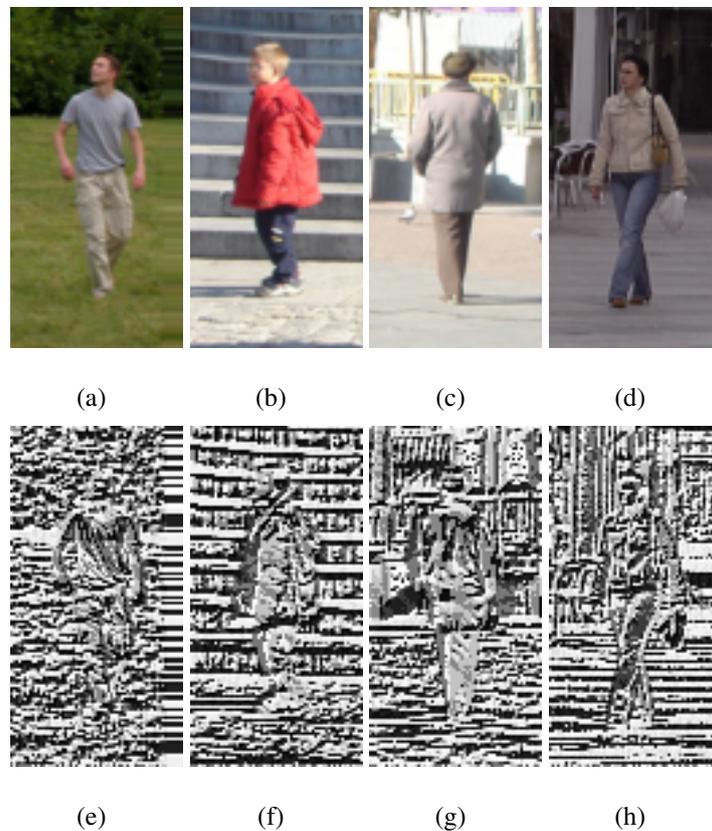


Figura 2.5: Imagens de pedestres e suas respectivas imagens LBP.

exemplo, o padrão 11111111 é considerado uniforme pois possui 0 transições. Os padrões 11110011 e 00111100 também são considerados uniformes, pois possuem 2 transições, enquanto os padrões 10001100 e 00110011 não são uniformes, pois possuem 4 transições cada. A partir de todas as combinações possíveis de um número binário de 8 dígitos, é possível for-

mar 58 padrões uniformes. Assim, o histograma é composto de 59 posições, onde 58 posições correspondem aos diferentes padrões uniformes e uma posição adicional é usada para atribuir todos os padrões não uniformes.

No HOG-LBP, esse histograma é calculado por bloco assim como acontece com o HOG. Para cada bloco, a combinação do HOG com LBP é feita simplesmente concatenando os resultados obtidos pelos dois descritores.

2.3.3 LTP

O *Linear Ternary Patterns* (LTP) [5] é uma generalização do LBP que é mais discriminante e menos sensível ao ruído em regiões uniformes. Quando há pequenas variações nas intensidades dos *pixels*, o LBP gera códigos diferentes, pois o limiar aplicado corresponde ao valor do *pixel* central da região 3x3. Já no caso do LTP, um intervalo é criado a partir de um limiar t e é aplicado à diferença entre o *pixel* central e cada vizinho, fazendo com que pequenas variações nas intensidades dos *pixels* não alterem o valor do código LTP. Como mostrado na equação 2.13, se a diferença estiver em um intervalo entre $-t$ e $+t$, é atribuído o código 0, caso seja menor ou igual a $-t$ é atribuído o código -1 e se for maior ou igual a $+t$, é atribuído o código 1, resultando em um código ternário. A Figura 2.6 mostra como é calculado o LTP para um limiar $t = 5$.

$$f(x) = \begin{cases} -1 & x \leq -t \\ 0 & ||x|| < t \\ 1 & x \geq t \end{cases} \quad (2.13)$$

2.3.4 LMEBP

O *Local Maximum Edge Binary Patterns* (LMEBP) [4] é calculado a partir da magnitude da diferença entre um *pixel* e seus vizinhos e extrai informação baseado na distribuição das bordas na imagem. Ela se diferencia do LBP, pois leva em consideração também a magnitude da borda,

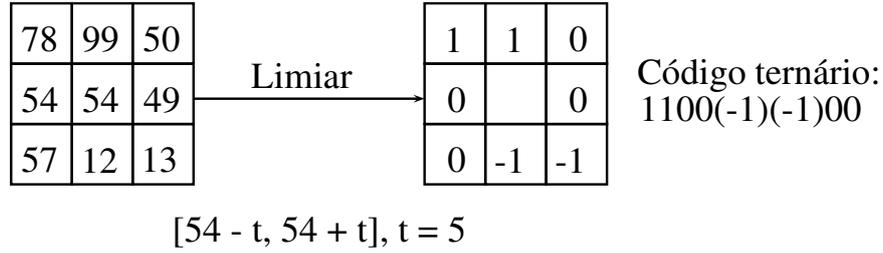


Figura 2.6: Extração do LTP de uma região 3x3 para um limiar igual a 5.

enquanto no LBP somente a direção é levada em consideração.

No LMEBP, a primeira maior borda é aquela cuja diferença entre o *pixel* central e seus vizinhos possui o maior valor absoluto. Considerando uma região 3x3, podemos aplicar a diferença entre o *pixel* central e seus vizinhos e encontrar a maior borda em valor absoluto a partir das equações 2.14 e 2.15, onde $\max(x)$ é a função que retorna o maior valor no vetor x .

$$I'(g_i) = I(g_i) - I(g_c) \quad i = 1, 2, \dots, 8 \quad (2.14)$$

$$i_1 = \arg(\max(|I'(g_1)|, |I'(g_2)|, |I'(g_3)|, \dots, |I'(g_8)|)) \quad (2.15)$$

Se a maior borda for positiva, é atribuído o valor 1 a esse *pixel* central, caso contrário, atribui-se 0, como se observa nas equações 2.16 e 2.17.

$$I^n(g_c) = f(I'(g_i)) \quad (2.16)$$

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{caso contrário} \end{cases} \quad (2.17)$$

E assim o LMEBP é definido como:

$$LMEBP(I(g_c)) = \{I^n(g_c), I^n(g_1), I^n(g_2), I^n(g_3), I^n(g_4), I^n(g_5), I^n(g_6), I^n(g_7), I^n(g_8)\} \quad (2.18)$$

Como cada posição corresponde ao valor 0 ou 1, o LMEBP(x) pode ser considerado um número binário de nove dígitos e convertido para decimal, resultando em um número entre

0 e 511. Depois de calcular o LMEBP para a 1ª maior borda em toda imagem, é possível construir um histograma que contém a frequência de ocorrências dos valores do LMEBP. Para uma imagem de tamanho $M \times N$ o histograma é calculado através da equação 2.19.

$$H_{\text{LMEBP}} = \sum_{j=1}^M \sum_{k=1}^N f(\text{LMEBP}(j,k), l), \in [0, 511] \quad (2.19)$$

$$f(x,y) = \begin{cases} 1 & x = y \\ 0 & \text{caso contrário} \end{cases} \quad (2.20)$$

Depois de calcular os LMEBPs para as segundas maiores bordas, terceiras maiores bordas e assim por diante, todos os histogramas são extraídos, de modo que obtemos 8 histogramas, cada um relacionado com um tipo de borda diferente.

Imagem				
2	5	3	9	1
6	7 ₍₆₎	6 ₍₇₎	1 ₍₈₎	5
2	3 ₍₅₎	6 ₍₀₎	8 ₍₁₎	2
3	8 ₍₄₎	4 ₍₃₎	2 ₍₂₎	9
1	2	3	2	1

Diferenças Locais		
1 g	0 h	-5 a
-3 c		2 f
2 e	-2 d	-4 b

Figura 2.7: Exemplo do cálculo do LMEBP em uma região 3×3 . Na esquerda, o *pixel* da região central com valor 6 e seus vizinhos. À direita, a diferença entre os vizinhos e o *pixel* central. Adaptado de [4].

Para ilustrar o cálculo do LMEBP, a Figura 2.7 mostra uma região de uma imagem arbitrária. Para o *pixel* central que possui valor 6 e se encontra na posição (0), calculamos a 1ª até a 8ª maior borda subtraindo-o dos seus vizinhos, e organizando o resultado em ordem decrescente de magnitude. Nesse caso, temos as bordas calculadas como mostra a Figura 2.8, ordenadas como -5, -4, -3, -2, 2, 2, 1, 0. Codificando esses valores em 0 para números menores que zero e 1 para os maiores ou iguais a zero, obtemos 0, 0, 0, 0, 1, 1, 1, 1. Os mesmos passos são realizados para os vizinhos, resultando na tabela da Figura 2.9.

a	b	c	d	e	f	g	h
-5	-4	-3	-2	2	2	1	0
0	0	0	0	1	1	1	1

Figura 2.8: Bordas ordenadas em função da sua magnitude e a codificação binária correspondente, atribuindo 0 para bordas negativas e 1 para bordas positivas [4].

	0	1	2	3	4	5	6	7	8	LMEBP	
a	0	0	1	1	0	1	0	0	1	105	1° LMEBP
b	0	0	1	1	0	1	0	1	1	107	2° LMEBP
c	0	0	1	1	0	1	0	0	1	105	3° LMEBP
d	0	0	1	0	0	1	0	0	1	073	4° LMEBP
e	1	0	1	0	0	1	0	1	1	331	5° LMEBP
f	1	0	0	0	0	1	0	1	1	267	6° LMEBP
g	1	0	1	0	0	0	0	0	1	321	7° LMEBP
h	1	1	1	0	0	1	0	1	1	459	8° LMEBP

Figura 2.9: Resultado do cálculo do LMEBP [4].

Na Figura 2.10, a imagem (b) foi calculada com o LBP, enquanto as demais representam da 1ª à 4ª maior borda do LMEBP. Como o LMEBP é ordenado em função da magnitude da borda, a imagem representada pela primeira maior borda do LMEBP possui um nível de detalhe muito maior do que as demais. A região compreendida pelo quadrado vermelho mostra a diferença entre o resultado obtido pelo LMEBP e pelo LBP.

2.4 Classificação

Os extratores de características HOG e HOG-LBP utilizam Máquinas de Vetores de Suporte (*Support Vector Machine* - SVM) [1, 10, 11] para classificar as características extraídas dos pedestres e não pedestres. O SVM é uma técnica supervisionada de aprendizagem de máquina, isto é, pares de entrada e saída desejada são apresentados durante a fase de treinamento, para que o classificador adquira capacidade de generalização, classificando corretamente entradas

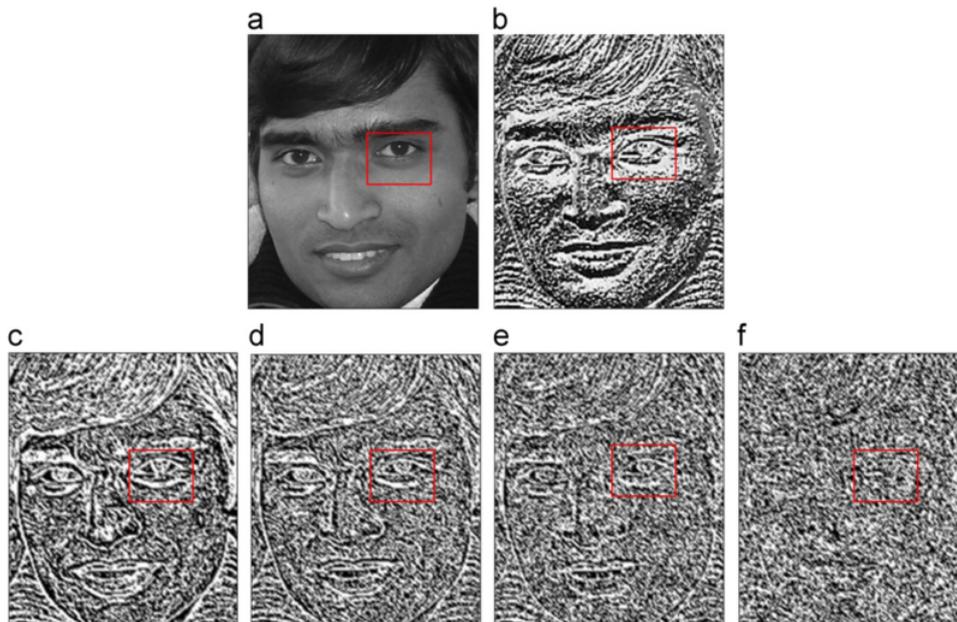


Figura 2.10: A figura (b) mostra a imagem LBP extraída da figura (a) e as figuras de (c) à (f) mostram as imagens obtidas a partir da 1ª maior borda até a 4ª maior borda do LMEBP [4].

não conhecidas.

Ao plotar os vetores do conjunto de treinamento em um espaço n-dimensional, é possível encontrar infinitas regiões que separam as duas classes de dados. No entanto, a função do SVM é encontrar um hiperplano que separa as classes dos dados de treinamento com a maior margem possível. Para isso, uma função kernel é utilizada para que os dados sejam projetados em um espaço de dimensão mais alta, tornando-os linearmente separáveis.

A Figura 2.11 mostra um exemplo de separação de duas classes ao se aplicar o SVM. Assim como a reta tracejada, há infinitas outras retas que separam as duas classes, porém a reta vermelha encontrada pelo SVM possui a maior margem de separação entre as classes possível, fornecendo uma boa capacidade de generalização.

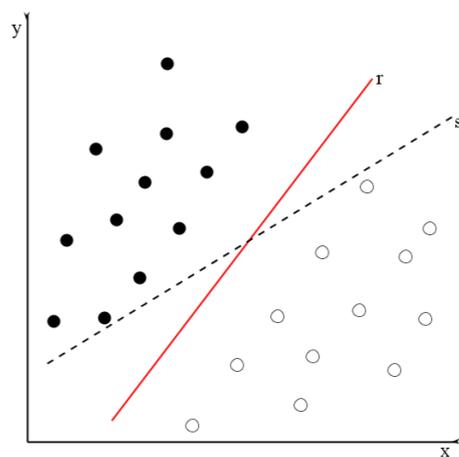


Figura 2.11: Exemplo da separação de duas classes utilizando SVM. A reta r encontrada pelo SVM separa as classes com margem máxima, enquanto a reta tracejada s separa com uma margem pequena.

Método Proposto

Este capítulo trata de duas novas abordagens propostas nesse trabalho para a detecção de pedestres. A primeira delas é a combinação do HOG com o LTP que chamamos de HOG-LTP e a segunda é uma combinação do HOG com o LMEBP, o HOG-LMEBP. Além desses novos extratores de características, propomos a combinação de classificadores a fim de atingir uma melhora na taxa de reconhecimento de pedestres sob condições de oclusão. Por fim, apresentamos a base de dados degradada, criada para simular a oclusão de forma controlada sobre as imagens de pedestres.

3.1 Combinação de características

O primeiro tipo de combinação de características adotado foi a junção do HOG com o LTP, que chamamos de HOG-LTP. A abordagem utilizada para a combinação dessas características é semelhante a adotada pelo HOG-LBP. Os HOGs calculados em cada célula são concatenados com histogramas de padrões uniformes do LTP extraídos da mesma célula. Como o LTP gera um código ternário para cada *pixel*, os padrões uniformes são calculados a partir da separação do LTP em códigos LBP superior e inferior.

A Figura 3.1 mostra como dois códigos binários são gerados a partir do código ternário. O padrão superior é formado pelos números maiores ou iguais a zero, enquanto o inferior é formado pelos menores ou iguais a zero, em valor absoluto.

Outra combinação proposta foi a união das características HOG com as do LMEBP. O descritor do LMEBP possui tamanho 8×512 , pois é formado por 8 histogramas, cada um contendo 512 posições. Dessa forma, a combinação HOG-LMEBP é formada através da concatenação do HOG de cada célula da imagem com o descritor do LMEBP para a mesma célula.

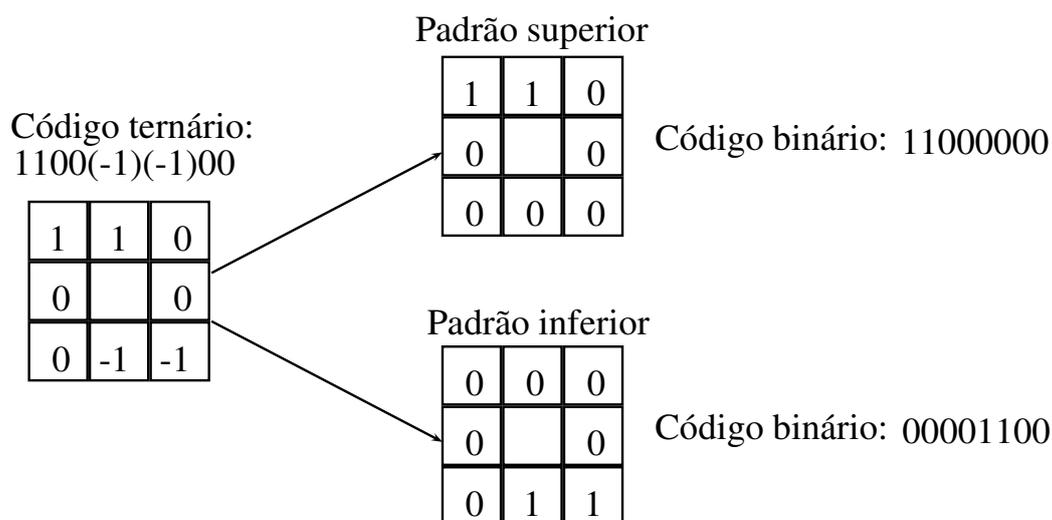


Figura 3.1: Separação do LTP em códigos LBP superior e inferior [5].

3.2 Combinação de classificadores

Em sistemas baseados em classificação, é comum testar o seu comportamento com diferentes características e classificadores. Para cada tipo de característica ou classificador, o sistema terá uma acurácia diferente, isto é, classificará corretamente um conjunto diferente de imagens. Assim também ocorre com os extratores de características utilizados neste trabalho. O HOG, HOG-LBP, HOG-LTP e HOG-LMEBP conseguem classificar corretamente a maior parte das imagens de pedestres e de não pedestres, porém algumas delas são classificadas corretamente por um classificador treinado com o HOG-LBP, por exemplo, mas não são classificadas corretamente pelo HOG-LTP. Verificando que esse comportamento também acontece sobre as imagens com oclusão sintética, decidimos combinar os classificadores [30, 31] de forma que todas as imagens classificadas corretamente pelos classificadores isolados continuem sendo corretamente classificadas no classificador combinado. Dessa forma, abordamos dois tipos de combinação de classificadores, sendo a primeira utilizando voto majoritário e a segunda através de um terceiro classificador SVM responsável por combinar os resultados dos dois primeiros.

3.2.1 Voto majoritário

A abordagem baseada em voto majoritário pode ser vista na Figura 3.2. Os classificadores SVM treinados cada um com a sua característica têm suas saídas combinadas, de forma que se pelo menos um dos dois classificar uma imagem como pedestre, o resultado final é que a imagem corresponde a um pedestre.

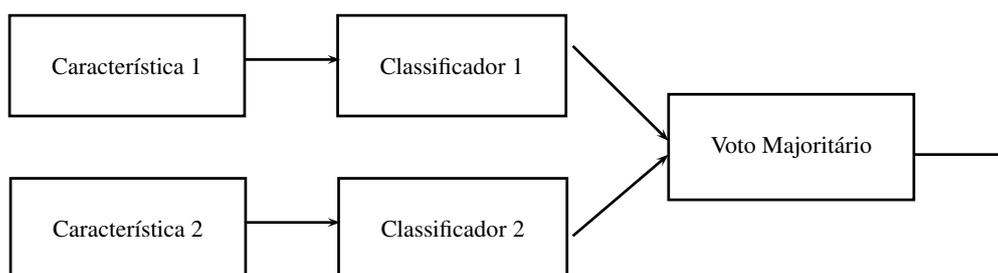


Figura 3.2: Combinação de classificadores utilizando voto majoritário.

Essa simples abordagem de combinação resultou em melhores taxas de detecção em condições de oclusão, que podem ser observadas no capítulo de experimentos.

3.2.2 Classificador em cascata

A segunda abordagem de combinação de classificadores é a de utilizar um terceiro classificador SVM para combinar os resultados dos dois primeiros. O esquema é mostrado na Figura 3.3. Assim como na abordagem por voto majoritário, os classificadores 1 e 2 são treinados com as suas características separadamente, porém, nesse caso, um classificador 3 combina os resultados obtidos pelos classificadores iniciais e decide qual o rótulo correto para a imagem cujas características foram extraídas.

Para treinar o classificador 3, passamos a utilizar o nível de confiança como saída dos classificadores 1 e 2, no lugar de utilizar o rótulo da classe. Assim, o classificador 3 é treinado a partir dos níveis de confiança obtidos dos primeiros classificadores. Além disso, um conjunto de imagens de treinamento diferente das utilizadas para treinar os classificadores 1 e 2, foi usado para treinar o classificador 3.

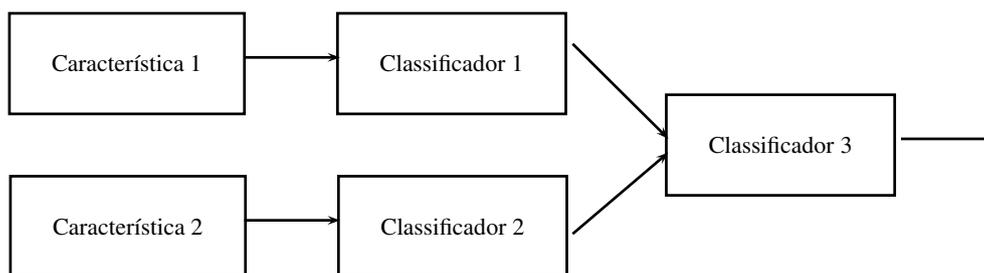


Figura 3.3: Combinação de classificadores utilizando classificador em cascata.

3.3 Base com oclusão sintética

A base de dados INRIA é bastante usada no treinamento e teste de sistemas de detecção de pedestres e, atualmente, a taxa de acerto sobre essa base é muito alta. Por outro lado, quando testado com imagens de pedestres sofrendo oclusão, a acurácia dos classificadores cai consideravelmente. A base de pedestres Caltech [26] contém imagens desafiadoras e com oclusão, porém ela não é controlada de modo que seja possível avaliar como a taxa de acerto de um classificador é afetada a medida que a oclusão aumenta e de acordo com a direção da oclusão. Diante disso, construímos uma nova base de imagens a partir da base INRIA adicionando oclusão sintética. O propósito dessa nova base de dados é de analisar o efeito de oclusões com tamanhos e direções diferentes sobre a taxa de acerto dos classificadores.

Todas as oclusões nessa nova base foram geradas mesclando imagens de pedestres da base INRIA com imagens de não pedestres. As imagens de não pedestres foram mescladas de várias formas, de modo a gerar oclusões nas direções ascendente, descendente e lateral (da esquerda para direita). A Figura 3.4 mostra exemplos dos 3 tipos de oclusão gerados.

Além disso, para cada direção, foram gerados conjuntos de imagens com tamanhos diferentes de oclusão. Para a ascendente e a descendente, 9 conjuntos com oclusão foram criados, cada um aumentando a oclusão em 8 *pixels* em relação ao anterior, de modo que o conjunto 1 possui uma área ocluída de 8 *pixels* de altura, o segundo 16 *pixels*, o terceiro 24 *pixels* e assim sucessivamente. Para a oclusão lateral, foram construídos 4 conjuntos, com oclusão de 8, 16, 24 e 32 *pixels* de largura. A Figura 3.5 mostra o exemplo da oclusão ascendente, onde a primeira imagem é a original da base INRIA e cada imagem subsequente pertence a um conjunto



Figura 3.4: Imagens da base criada com oclusão sintética. A imagem (a) possui oclusão ascendente, enquanto a (b) tem oclusão descendente e a (c) oclusão lateral.

com tamanho de oclusão diferente. É possível notar que na última imagem apenas a porção superior do corpo do pedestre é visível, enquanto a porção inferior se encontra completamente escondida.

Além dos conjuntos formados pelos 3 tipos de oclusão citados, um conjunto adicional misturado foi criado para cada direção de oclusão, formado por 50% de imagens obtidas igualmente de cada tamanho de oclusão e 50% de imagens sem oclusão. Esse conjunto reproduz melhor o caso real, pois é composto por oclusões de diferentes tamanhos e ele tem o objetivo de testar os classificadores quando submetido a diferentes tipos de oclusão.



Figura 3.5: A primeira imagem original da base INRIA e exemplos de imagens de pedestres dos 9 conjuntos formados pelas oclusões ascendentes.

Experimentos

Neste capítulo mostramos todos os experimentos executados utilizando as técnicas originais e as propostas, além de uma combinação das mesmas. Todos os experimentos são executados sobre a base INRIA original, como também sobre a base degradada. Tanto a base original como a degradada de pedestres é formada por imagens de 128×64 *pixels*, sendo 2172 imagens para treinar o classificador SVM e 1126 imagens de teste. Além das imagens de pedestres, 5000 imagens de não pedestres retiradas aleatoriamente de cenas da rua foram utilizadas para treinamento e 4000 para teste. Todo o código foi implementado em Matlab e para o classificador SVM foi utilizada a biblioteca LIBSVM [32].

Nos experimentos, a acurácia, ou taxa de correta aceitação, indica a taxa de acerto do classificador, isto é, a relação entre o total de imagens corretamente classificadas e o total de imagens da base de dados. Além da acurácia, também utilizamos curvas ROC (*Receiver Operating Characteristics*) para mostrar como os classificadores se comportam. As curvas ROC [33] são gráficos bidimensionais que mostram a relação entre as taxas de correta aceitação e de falsa aceitação em diferentes limiares de operação do classificador.

4.1 Experimentos na base original usando combinação de características

Os experimentos na base de dados original foram executados de forma que as imagens usadas no treinamento e teste dos classificadores não continham nenhuma oclusão. A Tabela 4.1 mostra os resultados obtidos com estes experimentos.

Nota-se que todas as técnicas, a menos do HOG-LMEBP, possuem alta acurácia, confirmando que a taxa de acerto sobre a base de dados INRIA é muito alta. Com melhor acurácia sobre as imagens de pedestres, o HOG-LBP possui a terceira maior taxa de acerto sobre ima-

Tabela 4.1: Resultados dos experimentos executados na base de dados original, sem oclusão.

Descritor	Acurácia Pedestre (%)	Acurácia não Pedestre (%)
HOG	95,20	98,25
HOG-LBP	97,86	96,67
HOG-LTP	96,89	98,55
HOG-LMEBP	93,25	87,02

gens de não pedestres. O HOG-LTP, uma das combinações de características propostas, obteve a segunda maior acurácia nas imagens de pedestres e a maior nas imagens de não pedestres. O HOG, obteve o terceiro melhor desempenho quando testado com imagens de pedestres, por outro lado teve um bom resultado nas imagens de não pedestres. O HOG-LMEBP, a outra combinação de características proposta, possui menor acurácia sobre as imagens de pedestres, como também a menor sobre as imagens de não pedestres. Analisando esses resultados, podemos perceber que a combinação do HOG com uma característica de textura contribuiu para a diminuição na taxa de acerto de imagens de não pedestres. Essa diminuição apenas não ocorreu no HOG-LTP, porém a diferença foi de apenas 0,20% comparado ao HOG. Isso pode estar relacionado com a importância que a silhueta dos pedestres tem para a sua detecção. O contorno da região das pernas, ombros e cabeça são características que ficam muito evidentes quando as bordas horizontais e verticais dos pedestres são extraídas através dos filtros do HOG e são muito importantes para diferenciar de imagens de não pedestres. Ao adicionar informações de textura, a acurácia sobre imagens de pedestres aumenta, porém acaba por diminuir a acurácia em imagens de não pedestres, visto que a textura é muito mais diversificada do que a silhueta dos pedestres.

As curvas ROC mostradas nas Figuras 4.1, 4.2, 4.3 e 4.4 confirmam que o desempenho do HOG-LMEBP foi o pior, enquanto mostra que o HOG-LTP obteve um desempenho melhor que as demais técnicas, sendo levemente superior ao HOG-LBP.

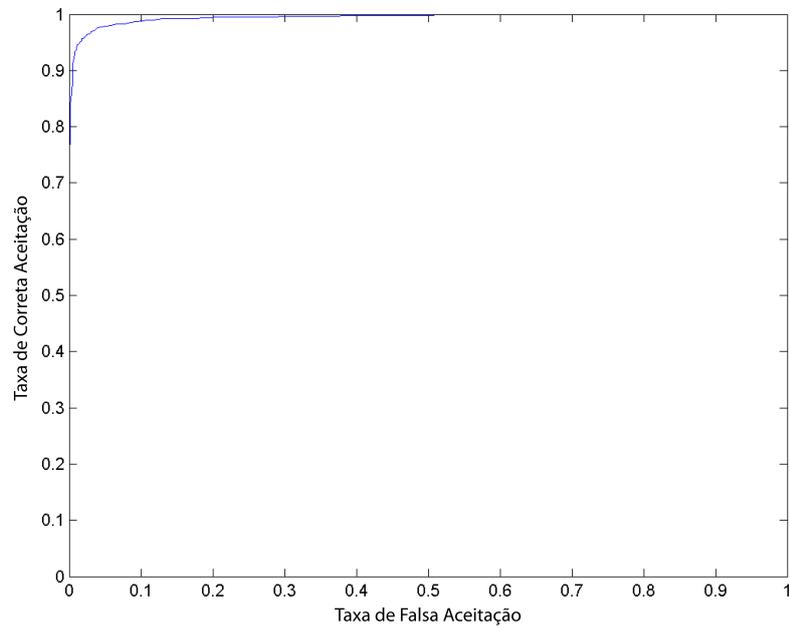


Figura 4.1: Curva ROC obtida para o HOG quando treinado e testado com imagens sem oclusão.

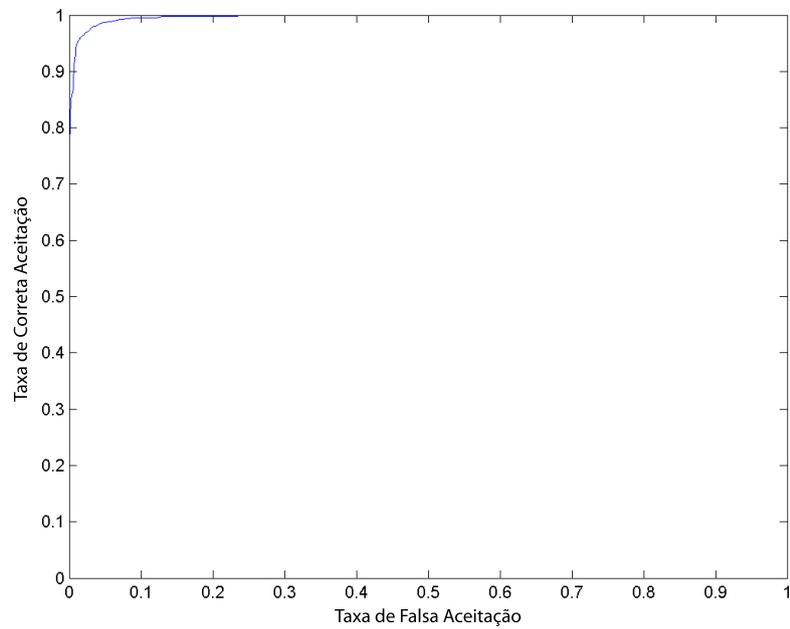


Figura 4.2: Curva ROC obtida para o HOG-LBP quando treinado e testado com imagens sem oclusão.

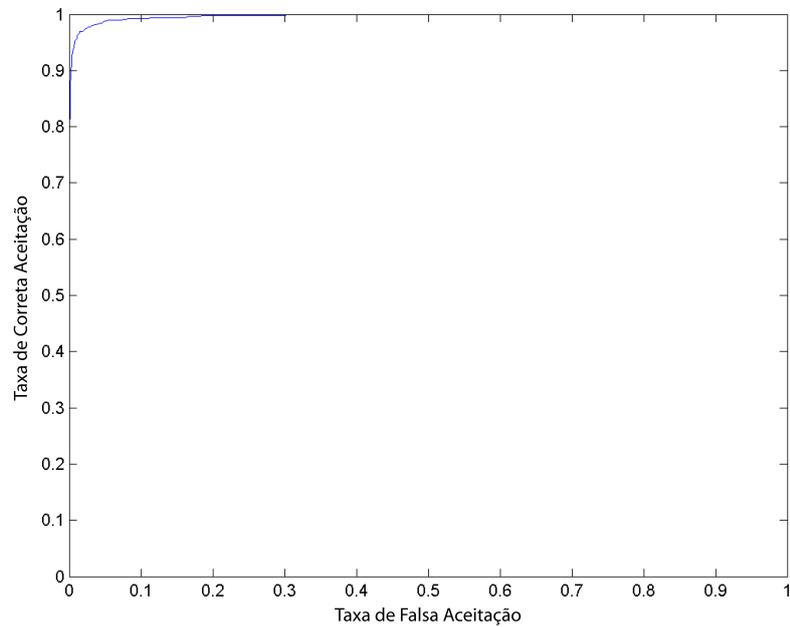


Figura 4.3: Curva ROC obtida para o HOG-LTP quando treinado e testado com imagens sem oclusão.

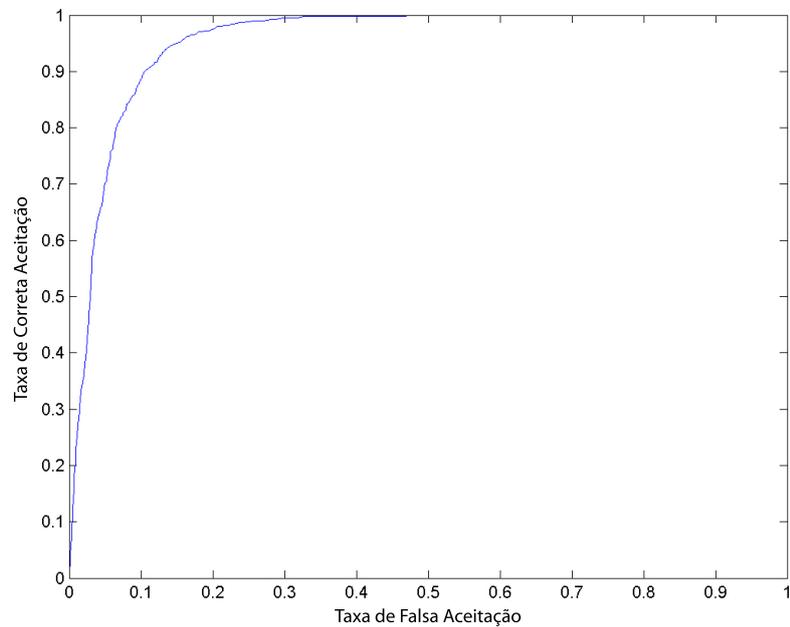


Figura 4.4: Curva ROC obtida para o HOG-LMEBP quando treinado e testado com imagens sem oclusão.

4.2 Experimentos na base degradada usando combinação de características

Para a base degradada, os experimentos foram realizados treinando os classificadores SVM com imagens sem oclusão e testando os mesmos sobre imagens com oclusão. O intuito desses experimentos é verificar como cada classificador se comporta a medida que a área de oclusão aumenta e, conseqüentemente, determinadas regiões do pedestre não ficam visíveis.

4.2.1 Oclusão ascendente

No conjunto de imagens com oclusão ascendente, a área de oclusão cresce de baixo para cima, escondendo primeiramente os pés e pernas, até esconder pouco mais da metade do corpo do pedestre. A Tabela 4.2 mostra os resultados ao testar os classificadores sobre os diversos tamanhos de oclusão.

Tabela 4.2: Resultados dos experimentos executados na base de dados formada pelos conjuntos de oclusão ascendente.

Área de Oclusão (%)	HOG (%)	HOG-LBP (%)	HOG-LTP (%)	HOG-LMEBP (%)
0,00	95,20	97,86	96,89	93,25
6,25	92,00	96,71	93,42	88,54
12,50	90,40	95,55	90,76	79,92
18,75	90,14	87,21	89,43	69,80
25,00	79,30	64,83	83,12	61,54
31,25	64,38	58,79	73,53	50,26
37,50	49,37	53,37	59,32	40,67
43,75	51,50	42,98	61,27	29,66
50,00	32,77	26,90	41,56	23,26
56,25	27,88	22,20	35,07	17,31

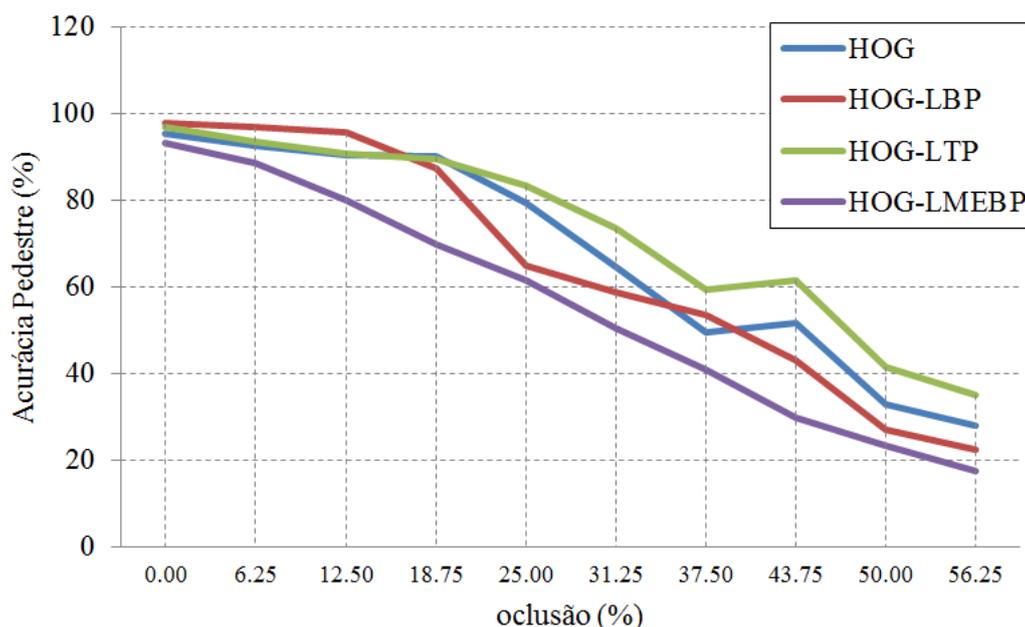


Figura 4.5: Comparativo das 4 técnicas quando submetidas à oclusão ascendente.

Através dos gráficos da Figura 4.5, é possível verificar como a taxa de acerto sobre imagens de pedestres diminui a medida que a região ocluída aumenta. O gráfico confirma que, apesar de apresentarem resultados próximos quando não existe oclusão, a medida que a oclusão aumenta, a acurácia diminui de forma diferente para cada técnica. Ao passar de 43%, há uma queda brusca na acurácia, motivada pela oclusão da metade do corpo e algumas vezes das mãos do pedestre.

Apesar do HOG-LBP possuir a maior taxa de acerto para imagens sem oclusão, a sua acurácia diminui significativamente por volta de 18% de região ocluída, ficando a frente apenas do HOG-LMEBP. O HOG-LTP se mostrou menos sensível à oclusão e obteve os melhores resultados a medida que a oclusão aumenta.

4.2.2 Oclusão descendente

Na oclusão descendente, a região de oclusão cresce de cima para baixo, escondendo primeiramente a cabeça, depois os ombros e braços do pedestre, até cobrir todo o tronco do pedestre, porém as suas pernas permanecem visíveis. A Tabela 4.3 mostra os resultados obtidos ao sub-

meter os classificadores baseados nos 4 extratores de características aos conjuntos com oclusão descendente.

Tabela 4.3: Resultados dos experimentos executados na base de dados formada pelos conjuntos de oclusão descendente.

Área de Oclusão (%)	HOG (%)	HOG-LBP (%)	HOG-LTP (%)	HOG-LMEBP (%)
0,00	95,20	97,86	96,89	93,25
6,25	95,55	98,22	96,09	85,25
12,50	95,91	98,40	94,84	72,91
18,75	93,42	91,65	92,54	66,16
25,00	77,88	68,38	71,84	59,50
31,25	62,34	54,70	61,54	53,55
37,50	52,75	43,42	50,08	45,64
43,75	54,79	44,49	50,44	39,34
50,00	46,53	34,19	40,31	33,21
56,25	42,80	29,21	38,54	26,37

Pelo gráfico da Figura 4.6 é possível verificar como se comporta a acurácia com o aumento da oclusão descendente. De modo semelhante ao ocorrido com a oclusão ascendente, o HOG-LBP tem melhor performance sobre condições normais, porém o HOG e o HOG-LTP o supera sobre condições de oclusão. Outro ponto interessante é que nós podemos confirmar a importância das características da cabeça e dos ombros para os classificadores. Após 18% de oclusão, a cabeça começa a ficar escondida e fica completamente coberta depois de 25% de oclusão. O gráfico mostra uma grande queda na acurácia após esse ponto para todos os classificadores.

4.2.3 Oclusão lateral

No tipo de oclusão lateral, a região de oclusão avança da esquerda para a direita do pedestre. Nesse caso, a oclusão se estende até a metade do seu corpo, escondendo primeiro um braço até

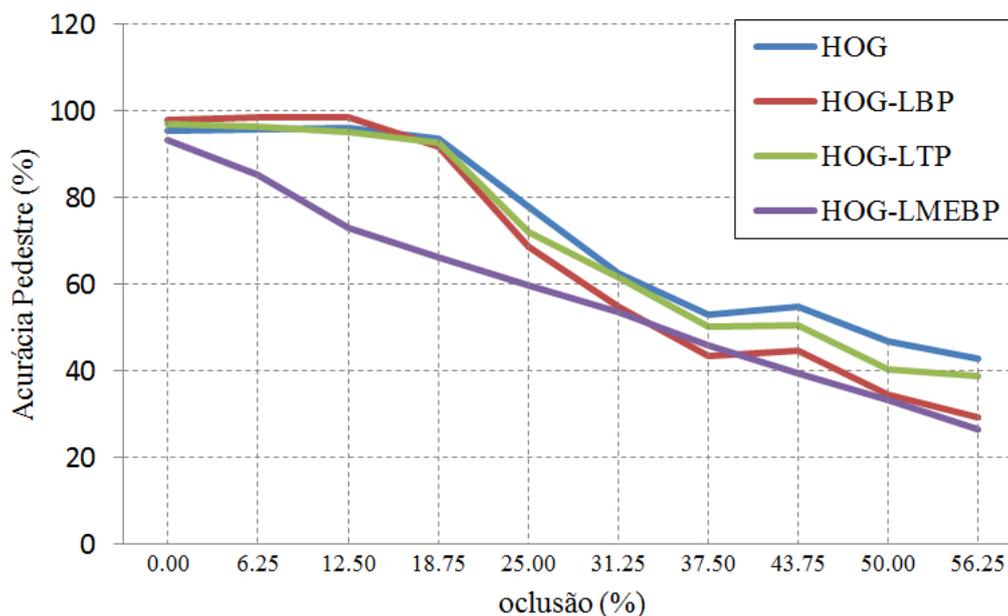


Figura 4.6: Comparativo das 4 técnicas quando submetidas à oclusão descendente.

ocluir parte da cabeça e das pernas.

Tabela 4.4: Resultados dos experimentos executados na base de dados formada pelos conjuntos de oclusão lateral.

Área de Oclusão (%)	HOG (%)	HOG-LBP (%)	HOG-LTP (%)	HOG-LMEBP (%)
0,00	95,20	97,86	96,89	93,25
12,50	94,04	97,42	95,91	80,28
25,00	91,65	94,84	93,78	59,23
37,50	81,52	71,75	84,01	43,96

Nos nossos experimentos, a oclusão do tipo lateral foi a que se mostrou menos prejudicial para a detecção de pedestres. O gráfico da Figura 4.7 mostra que, enquanto nos outros tipos de direção de oclusão a acurácia é menor que 60% para 37% de área ocluída, a oclusão lateral ainda permanece com taxa de acerto acima de 80% para o HOG e HOG-LTP. Isso indica que a simetria do pedestre garante que o lado do corpo não visível é amparado pelo outro lado. Também é possível observar que o HOG-LMEBP continuou com o pior resultado e o HOG

e HOG-LTP continuaram à frente do HOG-LBP, mesmo que com uma diferença menor na acurácia dos dois.

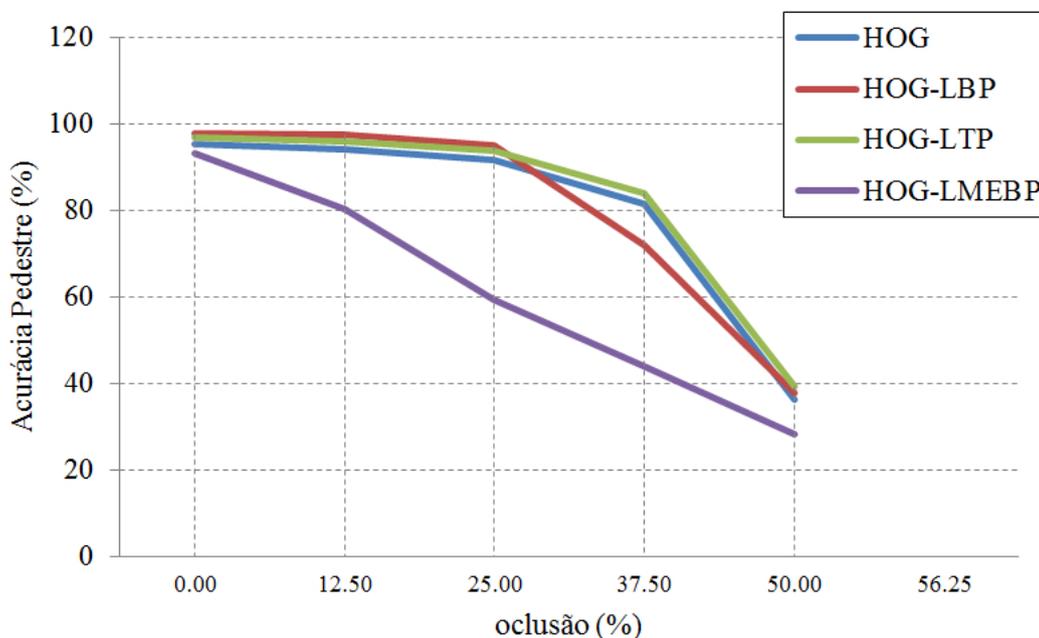


Figura 4.7: Comparativo das 4 técnicas quando submetidas à oclusão lateral.

4.2.4 Oclusão misturada

O último experimento realizado combinando-se características foi nos conjuntos misturados. Como citado anteriormente, esses conjuntos misturam 50% de imagens com oclusão com 50% de imagens sem oclusão. Sendo assim, foram formados 3 conjuntos misturados, cada um para um tipo de orientação de oclusão diferente. Estes conjuntos tentam se aproximar da situação real, visto que nesse caso, nem todos os pedestres estariam sofrendo oclusão ao mesmo tempo e as oclusões poderiam ser de diversos tamanhos. A Tabela 4.5 mostra o resultado dos testes realizados sobre esses conjuntos.

Da mesma forma que aconteceu com os outros conjuntos, o HOG-LTP superou as outras 3 características. Como nesses conjuntos há muitas imagens sem oclusão, o resultado do HOG-LBP se aproximou do HOG.

Tabela 4.5: Resultados dos experimentos executados nas bases de dados formadas por imagens sem oclusão misturadas com imagens com oclusões de diferentes tamanhos.

Tipo de Oclusão	HOG (%)	HOG-LBP (%)	HOG-LTP (%)	HOG-LMEBP (%)
Ascendente	80,63	80,81	83,21	70,33
Descendente	82,32	81,61	82,80	72,38
Lateral	85,34	85,87	88,27	72,02

4.3 Experimentos na base degradada usando combinação de classificadores

Após analisar os resultados obtidos com a combinação de características, percebemos que o HOG-LTP obteve melhores resultados sobre imagens com oclusão e, em alguns casos, os resultados foram muito próximos ao HOG. Além disso, cada classificador formado pelo HOG, HOG-LBP ou HOG-LTP detectou corretamente alguns pedestres ocluídos que outros classificadores não conseguiram detectar. Baseado nessas análises, nós combinamos o classificador HOG-LTP com o classificador HOG e também com o HOG-LBP. O classificador HOG-LMEBP não foi usado em combinação com os demais, pois ele não obteve bons resultados na maioria das situações de oclusão.

A primeira abordagem de combinação dos classificadores foi de classificar como um pedestre quando qualquer um dos classificadores determina que a imagem é de um pedestre. A segunda abordagem consistiu em usar um classificador SVM para combinar o resultado dos dois classificadores. A última abordagem não levou a bons resultados, enquanto a primeira melhorou a acurácia significativamente.

4.3.1 Oclusão ascendente

O primeiro experimento com a combinação de classificadores foi na base com oclusões ascendentes. A Tabela 4.6 mostra o resultado da combinação dos classificadores HOG + HOG-LTP

e HOG-LBP + HOG-LTP.

Tabela 4.6: Resultados dos experimentos executados na base com oclusões ascendentes utilizando combinação de classificadores.

Área de Oclusão (%)	HOG + HOG-LTP (%)	HOG-LBP + HOG-LTP (%)
0,00	97,51	98,85
6,25	96,45	98,31
12,50	94,05	96,89
18,75	93,52	95,74
25,00	88,10	88,99
31,25	78,42	81,71
37,50	65,36	70,60
43,75	67,85	70,69
50,00	48,49	49,47
56,25	41,21	42,36

O gráfico da Figura 4.8 mostra claramente que os resultados das combinações melhoram a acurácia sobre os classificadores isolados. É interessante notar que as curvas do gráfico para os classificadores combinados seguem um formato similar à dos classificadores isolados.

4.3.2 Oclusão descendente

A combinação de classificadores também obteve melhores resultados do que os classificadores isolados para os conjuntos com oclusão descendente. O HOG e a combinação HOG-LBP + HOG-LTP obtiveram resultados semelhantes, enquanto a combinação HOG + HOG-LTP se saiu um pouco melhor. A Tabela 4.7 e o gráfico da Figura 4.9 mostram os resultados dessa combinação.

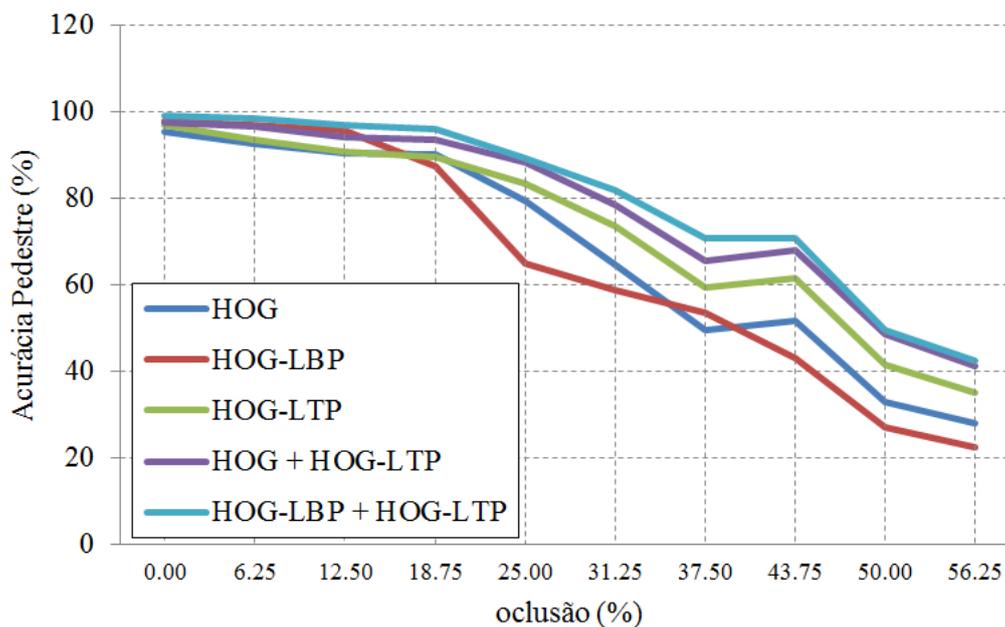


Figura 4.8: Comparativo entre as 3 técnicas isoladas e combinadas quando submetidas à oclusão ascendente.

Tabela 4.7: Resultados dos experimentos executados na base com oclusões descendentes utilizando combinação de classificadores.

Área de Oclusão (%)	HOG + HOG-LTP (%)	HOG-LBP + HOG-LTP (%)
0,00	97,51	98,85
6,25	97,60	98,49
12,50	97,69	98,58
18,75	95,12	96,36
25,00	81,35	77,89
31,25	67,67	65,99
37,50	56,75	55,15
43,75	58,35	55,06
50,00	49,91	45,83
56,25	47,25	43,07

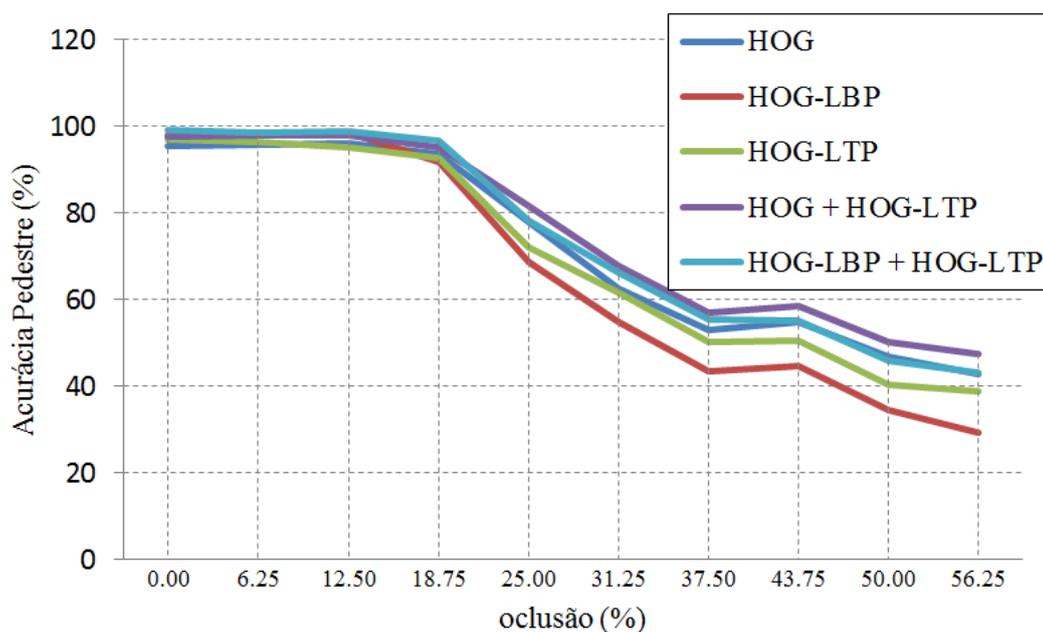


Figura 4.9: Comparativo entre as 3 técnicas isoladas e combinadas quando submetidas à oclusão descendente.

4.3.3 Oclusão lateral

Para as oclusões laterais, a combinação dos classificadores também mostrou melhores resultados que os classificadores isolados. Dessa vez, a acurácia da combinação HOG-LBP + HOG-LTP foi ligeiramente superior que a combinação HOG + HOG-LTP. A Tabela 4.8 e o gráfico da Figura 4.10 mostram os resultados.

4.3.4 Oclusão misturada

Como esperado, o resultado da combinação de classificadores sobre o conjunto misturado também obteve acurácia superior a dos classificadores isolados. A Tabela 4.9 mostra que para todos os tipos de direção de oclusão, a combinação HOG-LBP + HOG-LTP obteve a melhor performance. Isso indica que a combinação desses classificadores também deve obter resultados melhores em cenários reais, pois, nesse caso, estão presentes pedestres ocluídos e não ocluídos.

Tabela 4.8: Resultados dos experimentos executados na base com oclusões laterais utilizando combinação de classificadores.

Área de Oclusão (%)	HOG + HOG-LTP (%)	HOG-LBP + HOG-LTP (%)
0,00	97,51	98,85
12,50	97,25	98,58
25,00	95,91	97,87
37,50	88,37	89,70
50,00	45,74	50,00

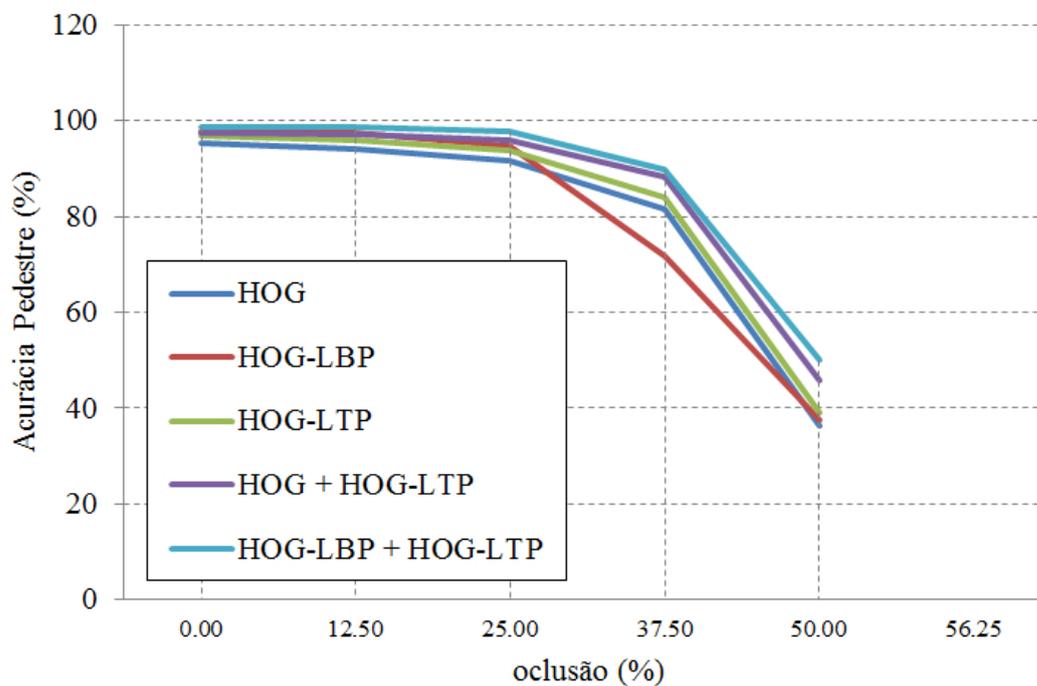


Figura 4.10: Comparativo entre as 3 técnicas isoladas e combinadas quando submetidas à oclusão lateral.

Tabela 4.9: Resultados dos experimentos executados sobre o conjunto misturado utilizando combinação de classificadores.

Tipo de Oclusão	HOG + HOG-LTP (%)	HOG-LBP + HOG-LTP (%)
Acidente	86,50	88,72
Descendente	85,79	86,15
Lateral	90,05	92,01

4.3.5 Imagens de não pedestres

A combinação de classificadores foi realizada adotando a imagem como sendo de pedestre caso qualquer um dos dois classificadores classifique-a como pedestre. É esperado que essa abordagem interfira na acurácia sobre as imagens de não pedestres. Isso acontece porque mesmo que os classificadores discordem quanto a classificação, a imagem continuará sendo classificada como pedestre, quando poderia não ser. Sendo assim, realizamos testes com os classificadores combinados sobre o conjunto de não pedestres e comparamos com os resultados obtidos para os classificadores isolados.

A abordagem de combinação de classificadores acarretou uma pequena diminuição nas taxas de detecção de imagens de não pedestres. Apesar disso, o aumento significativo na acurácia de detecção de pedestres justificam a aplicação do método combinado. As Tabelas 4.10 e 4.11 mostram os resultados para os classificadores isolados e combinados.

Tabela 4.10: Resultados dos experimentos executados com os classificadores isolados sobre o conjunto de imagens de não pedestres.

	HOG (%)	HOG-LBP (%)	HOG-LTP (%)	HOG-LMEBP (%)
Acurácia	98,25	96,67	98,55	87,02

Para a combinação HOG + HOG-LTP, a acurácia diminuiu apenas 1,1% comparado com o HOG e 1,4% quando comparado ao HOG-LTP. No caso da combinação HOG-LBP + HOG-

Tabela 4.11: Resultados dos experimentos executados com os classificadores combinados sobre o conjunto de imagens de não pedestres.

	HOG + HOG-LTP (%)	HOG-LBP + HOG-LTP (%)
Acurácia	97,15	95,37

LTP, a queda na acurácia foi um pouco maior, sendo de 1,3% comparado ao HOG-LBP e 3,18% comparado ao HOG-LTP.

Conclusões e Trabalhos Futuros

Este trabalho mostrou a importância da detecção de pedestres, onde algumas das suas aplicações incluem as áreas de sistemas de vigilância, automação veicular e robótica. Apesar de muito ter se avançado nessa área, ainda há muito o que se aprimorar. Em imagens sem oclusão, as técnicas de detecção de pedestres funcionam com uma taxa de acerto próxima de 100%, porém, quando a oclusão é adicionada, essa taxa pode cair bruscamente. Sendo assim, estudos foram realizados para entender como a detecção de pedestres é afetada sob condições de oclusão. Através da base com oclusão sintética, foi possível analisar como a acurácia é afetada quando determinadas regiões do pedestre estão ocluídas. Além disso, foi possível estudar e propor novas técnicas de detecção.

A detecção de pedestres foi descrita em etapas, sendo elas: pré-processamento, extração de características e classificação. O funcionamento dos classificadores utilizando HOG e HOG-LBP foram descritos segundo estas etapas, sendo o HOG utilizado como base para os classificadores da abordagem proposta. A utilização do HOG se deu porque ele é um classificador bastante usado como base de outras técnicas.

No método proposto, foram construídos dois novos extratores de características em combinação com o HOG, sendo eles o HOG-LTP e o HOG-LMEBP. Além disso, foram propostas combinações de classificadores em cascata e através do voto majoritário. Também foi mostrado como a base com oclusão sintética, que possui oclusões em várias direções e tamanhos, foi construída.

Finalmente, na seção de experimentos, mostramos os resultados dos extratores de características e da combinação de classificadores propostos quando testados sobre as bases com e sem oclusão sintética. Os resultados mostraram que o HOG-LTP e que a combinação de classificadores superam os resultados dos classificadores originais quando submetidos à oclusão.

Os experimentos também confirmaram que a oclusão degrada bastante a acurácia dos clas-

sificadores. Quando ocluídos, a cabeça e os ombros foram grandes responsáveis pela queda no desempenho. Outro ponto importante é que a simetria do pedestre contribui para que a queda na acurácia devido à oclusão lateral seja pequena. Quanto ao desempenho computacional, o cálculo do HOG levou em média 8ms para ser executado por janela, enquanto o HOG-LMEBP 41ms, o HOG-LBP 70ms e o HOG-LTP 106ms. Como todos os algoritmos foram implementados em MATLAB, acredito que é possível melhorar consideravelmente esses tempos de execução, otimizando o código e utilizando uma linguagem mais rápida.

Para trabalhos futuros, estão a criação de outras bases com oclusão sintética para verificar como os novos classificadores se comportam com imagens de bases diferentes e também a análise de como outros tipos de ruídos afetam a detecção, como os causados por imagens com baixa resolução ou por problemas de iluminação.

Como fruto deste trabalho, um artigo intitulado *Pedestrian Detection Under Progressive Occlusion* foi publicado na *IEEE International Conference on Systems, Man, and Cybernetics, 2013 (SMC)*.

Referências Bibliográficas

- [1] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:886–893, 2005.
- [2] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing (3rd Edition)*. Prentice Hall, 2007.
- [3] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.
- [4] M. Subrahmanyam, R.P. Maheshwari, and R. Balasubramanian. Local maximum edge binary patterns: A new descriptor for image retrieval and object tracking. *Signal Processing*, 92(6):1467–1479, June 2012.
- [5] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 19(6):1635–50, June 2010.
- [6] Markus Enzweiler and Dariu M Gavrila. Monocular pedestrian detection: survey and experiments. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2179–95, December 2009.
- [7] David Gerónimo, Antonio M López, Angel D Sappa, Thorsten Graf, D Gerónimo, and A M López. Survey of pedestrian detection for advanced driver assistance systems. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1239–58, July 2010.

- [8] Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. New features and insights for pedestrian detection. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1030–1037, June 2010.
- [9] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, January 1996.
- [10] Xiaoyu Wang, Tony X. Han, and Shuicheng Yan. An HOG-LBP human detector with partial occlusion handling. *2009 IEEE 12th International Conference on Computer Vision*, pages 32–39, September 2009.
- [11] Guolong Gan and Jian Cheng. Pedestrian Detection Based on HOG-LBP Feature. *2011 Seventh International Conference on Computational Intelligence and Security*, pages 1184–1187, December 2011.
- [12] Luis Malagón-Borja and Olac Fuentes. Object detection using image reconstruction with PCA. *Image and Vision Computing*, 27(1-2):2–9, January 2009.
- [13] D.S. Bolme, B.A. Draper, and J.R. Beveridge. Average of Synthetic Exact Filters. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2105–2112. IEEE, June 2009.
- [14] David S Bolme, Yui Man Lui, Bruce A Draper, and J Ross Beveridge. Simple real-time human detection using a single correlation filter. In *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–8. IEEE, December 2009.
- [15] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–45, September 2010.

- [16] Bo Wu and Ram Nevatia. Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. *International Journal of Computer Vision*, 75(2):247–266, January 2007.
- [17] Payam Sabzmeydani and Greg Mori. Detecting Pedestrians by Learning Shapelet Features. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2007.
- [18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–511–I–518, 2001.
- [19] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
- [20] Greg Mori, Serge Belongie, and Jitendra Malik. Efficient shape matching using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1832–7, November 2005.
- [21] Christian Wojek and B Schiele. A performance evaluation of single and multi-feature people detection. *Pattern Recognition*, pages 82–91, 2008.
- [22] Piotr Dollar, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral Channel Features. In *Proceedings of the British Machine Vision Conference 2009*, pages 91.1–91.11. British Machine Vision Association, 2009.
- [23] Piotr Dollar, Serge Belongie, Pietro Perona, and Piotr Dollár. The Fastest Pedestrian Detector in the West. In *Proceedings of the British Machine Vision Conference 2010*, pages 68.1–68.11. British Machine Vision Association, 2010.
- [24] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S. Davis. Human detection using partial least squares analysis. In *2009 IEEE 12th International Conference on Computer Vision*, number Iccv, pages 24–31. IEEE, September 2009.

- [25] Ogê Marques Filho and Hugo Vieira Neto. *Processamento Digital de Imagens*, volume 21/03. Brasport, 1999.
- [26] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, July 2011.
- [27] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.
- [28] Timo Ahonen, Abdenour Hadid, Matti Pietik, and M Pietikäinen. Face Recognition with Local Binary Patterns. *Computer Vision ECCV 2004*, 3021:469–481, 2004.
- [29] T Ojala, M Pietikainen, and T Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.
- [30] Hyunjoong Kim, Hyeuk Kim, Hojin Moon, and Hongshik Ahn. A weight-adjusted voting algorithm for ensembles of classifiers. *Journal of the Korean Statistical Society*, 40(4):437–449, December 2011.
- [31] L Lam and S Y Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance, 1997.
- [32] Chih-chung Chang and Chih-jen Lin. LIBSVM: a library for support vector machines. *Computer*, 2(3):1–30, 2001.
- [33] Tom Fawcett. An introduction to ROC analysis. *Pattern Recogn Lett*, 27(8):861–874, 2006.