



DISSERTAÇÃO DE MESTRADO

**UMA NOVA METODOLOGIA PARA CATEGORIZAÇÃO DE VEÍCULOS
TERRESTRES BASEADA EM MOMENTOS E MÉTRICAS
DE DESEMPENHO E COMPLEXIDADE COMPUTACIONAL**

Ticiano Augusto Callai Bragatto

Brasília, julho de 2009

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

**UMA NOVA METODOLOGIA PARA CATEGORIZAÇÃO DE VEÍCULOS
TERRESTRES BASEADA EM MOMENTOS E MÉTRICAS
DE DESEMPENHO E COMPLEXIDADE COMPUTACIONAL**

Ticiano Augusto Callai Bragatto

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Mestre em Engenharia Elétrica*

Banca Examinadora

Prof. Marcus Vinicius Lamar, CiC / UnB

Orientador

Prof. Aldebaro Barreto da Rocha Klautau Junior,

ENGCOMP / UFPA
Examinador externo

Prof. Alexandre Romariz, ENE / UnB

Examinador interno

FICHA CATALOGRÁFICA

BRAGATTO, TICIANO AUGUSTO CALLAI

Uma Nova Metodologia para Categorização de Veículos Terrestres Baseada em Momentos e Métricas de Desempenho e Complexidade Computacional [Distrito Federal] 2009.

ix, 51 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2009)

Dissertação de Mestrado - Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

- | | |
|-----------------------------------|------------------------------|
| 1. Visão Computacional | 2. Categorização de Veículos |
| 3. Momentos | 4. Redes Neurais Artificiais |
| 5. Máquinas de Vetores de Suporte | |

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

BRAGATTO, T. A. C. (2009). Uma Nova Metodologia para Categorização de Veículos Terrestres Baseada em Momentos e Métricas de Desempenho e Complexidade Computacional. Dissertação de Mestrado, Publicação PPGENE.DM - 385/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 51 p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Ticiano Augusto Callai Bragatto

TÍTULO DA MONOGRAFIA DE PROJETO FINAL: Uma Nova Metodologia para Categorização de Veículos Terrestres Baseada em Momentos e Métricas de Desempenho e Complexidade Computacional.

GRAU / ANO: Mestre em Engenharia Elétrica / 2009

É concedida à Universidade de Brasília a permissão para reproduzir cópias desta monografia de Projeto Final e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta monografia de Projeto Final pode ser reproduzida sem a autorização por escrito do autor.

Ticiano Augusto Callai Bragatto

CEFTRU, Campus Darcy Ribeiro

70.919-970 - Brasília/DF - Brasil

Dedicatória

Aos meus pais, por terem me dado a melhor criação possível, à minha irmã, por inúmeras discussões produtivas e à Andressa, que, apesar da distância, continua sendo uma excelente companheira.

Ticiano Augusto Callai Bragatto

Agradecimentos

Aos membros da banca, por suas contribuições e ao meu orientador, por suportar um orientando inquieto ao longo de 8 anos de pesquisa.

Aos meus familiares e amigos, por fazerem da minha vida algo tão divertido e produtivo, mesmo que no sentido mais ocioso possível.

No CEFTRU, aos meus colegas de equipe Gabriel e Augusto e aos colegas Daniel e George, por inúmeras contribuições em um tema que não é de meu pleno entendimento: Transportes.

Por último, a Dietrich Mateschitz, que em uma viagem de negócios à Tailândia, em 1982, resolveu ocidentalizar uma receita, abrindo uma das maiores companhias de bebidas do mundo. Sem ele, não existiria Red Bull e sem essa bebida, esta dissertação provavelmente não seria concluída a tempo.

Ticiano Augusto Callai Bragatto

RESUMO

Este trabalho propõe o uso de momentos para executar a categorização de veículos através de processamento de imagens. Foram extraídos momentos principais e momentos invariantes à escala e à translação de um total de 528 imagens, divididas uniformemente entre as seguintes classes: carros, vans, ônibus e caminhões. Essas características foram ordenadas pela sua significância para a classificação de veículos nas quatro classes já apresentadas. Utilizando os cinco conjuntos de características que obtiveram as melhores taxas médias de acerto, classificadores como k -NN, TM , MLP, AVA-SVM, OVA-SVM e PT-SVM, foram utilizados para definir qual a combinação entre os cinco conjuntos de características e os classificadores que obteve o melhor desempenho. As métricas analisadas consistem em taxa média de acerto, desvio padrão, custo computacional e FROTA - Frequência Relativa de Ocorrência de Taxa de Acerto, que fornece um instrumento baseado na distribuição das taxas de acerto e permite a análise visual da precisão e robustez de classificadores. A metodologia proposta atingiu taxas médias de acerto da ordem de 97% com desvio padrão de 1,27 para classificação utilizando MLP e um conjunto formado pelos 9 momentos invariantes à escala e à translação extraídos.

ABSTRACT

This study proposes using moments to categorize vehicles through image processing. Principal moment features and moment features that are invariant to scale and translation were extracted from a total of 528 images uniformly divided into the following classes: cars, vans, buses and trucks. These features are ordered according to their significance to classifying vehicles in the four classes mentioned above. Using the 5 sets of features that gave the best hit rates, classifiers such as k -NN, TM , MLP, AVA-SVM, OVA-SVM and PT-SVM are used to determine which combination between the sets of features and the classifiers has the best performance. The metrics analyzed in this study are based on hit rates, standard deviation, the cost of computing and the relative frequency of correct predictions (FROTA). The latter provides an instrument based on hit rates and allows for a visual analysis of the classifiers' accuracy and resistance. The methodology proposed in this study yielded a hit rate of 97% with a 1,27 standard deviation for classifications using MLP and 9 invariant moments.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	OBJETIVO E JUSTIFICATIVA	2
1.3	ORGANIZAÇÃO	3
2	REVISÃO BIBLIOGRÁFICA	5
2.1	CATEGORIZAÇÃO DE VEÍCULOS	5
2.2	CATEGORIZAÇÃO DE VEÍCULOS POR VISÃO COMPUTACIONAL	5
3	METODOLOGIA UTILIZADA	8
3.1	EXTRAÇÃO DE CARACTERÍSTICAS	8
3.1.1	MOMENTOS PRINCIPAIS	9
3.1.2	MOMENTOS INVARIANTES À TRANSLAÇÃO E AO ESCALONAMENTO	9
3.2	MEDIÇÃO DE DESEMPENHO	11
3.3	COMPLEXIDADE COMPUTACIONAL	14
3.4	SELEÇÃO DE CARACTERÍSTICAS	16
3.4.1	<i>SFFS – Sequential Floating Forward Selection</i>	16
3.4.2	<i>SFBS – Sequential Floating Backward Selection</i>	16
3.5	CLASSIFICADORES CONSIDERADOS	17
3.5.1	K-NN	17
3.5.2	<i>TM - Template Matching</i>	18
3.5.3	<i>MLP – REDES NEURAS ARTIFICIAIS Multi-Layer Perceptron</i>	19
3.5.4	MÁQUINAS DE VETORES DE SUPORTE	19
4	RESULTADOS OBTIDOS	24
4.1	EXTRAÇÃO DE CARACTERÍSTICAS	26
4.2	MEDIÇÃO DE DESEMPENHO E CONJUNTOS DE TREINO E TESTE	27
4.3	SELEÇÃO DE CARACTERÍSTICAS	27
4.4	<i>k-Nearest Neighbour</i>	30
4.4.1	$k = 1$	30
4.4.2	$k = 3$	31
4.5	<i>Template Matching</i>	32
4.6	MLP	36
4.7	<i>SVM – Support Vector Machines</i>	38
4.7.1	<i>AVA – All versus All</i>	38
4.7.2	<i>OVA – One versus All</i>	40
4.7.3	<i>PT-SVM</i>	41
5	CONCLUSÕES	47
5.1	TRABALHOS FUTUROS	47

LISTA DE FIGURAS

1.1	Esquemas de Contagem: a) Linha e b) Cordão	2
1.2	Exemplo de formulários utilizados em pesquisa de campo em [1]	3
1.3	Categorias de Veículos: a) Carro leve, b) van, c) ônibus e d) caminhão	4
2.1	Classes utilizadas em [2]: a) carro <i>sedan</i> , b) van, c) carro <i>hatch</i> , d) ônibus e e) caminhão.....	6
2.2	Classes utilizadas em [3]: a) carro <i>sedan</i> , b) van, c) camioneta, d) caminhão aberto, e) caminhão baú, f) ônibus e g) caminhão articulado.	6
3.1	Geração de Máscara: a) Imagem Original e b) Máscara Obtida	8
3.2	Exemplo de FROTA e Gaussiana.	12
3.3	Classificadores com precisão diferentes.....	13
3.4	Classificadores com robustez distintas.....	14
3.5	Exemplo de Classificador $k - NN$	18
3.6	Exemplo de Classificador TM	19
3.7	Exemplo de Rede Neural MLP.....	20
3.8	Exemplo de Hiper-planos de SVM.....	21
3.9	Aumento de Dimensão	21
3.10	Exemplo de Classificador PT-SVM	23
4.1	Mono-pé de suporte.....	24
4.2	Câmera e suporte utilizados.....	25
4.3	Imagens capturadas de diferentes pontos de vista	25
4.4	Programa para segmentação de veículos durante uso.....	26
4.5	Funções de densidade de probabilidade de acerto utilizadas na seleção de características	30
4.6	$k-NN$, euclidiana, $k = 1$ – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	31
4.7	$k-NN$, <i>city block</i> , $k = 1$ – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	32
4.8	$k-NN$, euclidiana, $k = 3$ – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	33
4.9	$k-NN$, <i>city block</i> , $k = 3$ – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	34
4.10	TM , euclidiana – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	35
4.11	TM , <i>city block</i> – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	36
4.12	MLP – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	37
4.13	AVA – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	39
4.14	OVA – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	41
4.15	Topologia da Árvore PT-SVM	42
4.16	PT-SVM – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5	45

LISTA DE TABELAS

3.1	Complexidade	15
4.1	Exemplo de Momentos Principais.....	26
4.2	Exemplo de Momentos Invariantes à Escala e Translação	27
4.3	Exemplo de Características Normalizadas	28
4.4	Resultado da Ordenação de Características utilizando <i>SFFS</i>	29
4.5	<i>k-NN</i> , euclidiana, $k = 1$ – Taxa Média de Acerto, Desvio Padrão e ICC	31
4.6	<i>k-NN</i> , <i>city block</i> , $k = 1$ – Taxa Média de Acerto, Desvio Padrão e ICC.....	31
4.7	<i>k-NN</i> , euclidiana, $k = 3$ – Taxa Média de Acerto, Desvio Padrão e ICC	32
4.8	<i>k-NN</i> , <i>city block</i> , $k = 3$ – Taxa Média de Acerto, Desvio Padrão e ICC.....	33
4.9	<i>TM</i> , euclidiana – Taxa Média de Acerto, Desvio Padrão e ICC	33
4.10	<i>TM</i> , <i>city block</i> – Taxa Média de Acerto, Desvio Padrão e ICC	34
4.11	MLP – Taxa Média de Acerto, Desvio Padrão e ICC.....	38
4.12	AVA–SVM – Taxa Média de Acerto, Desvio Padrão e ICC	38
4.13	OVA–SVM – Taxa Média de Acerto, Desvio Padrão e ICC	40
4.14	Tamanho da Frota Brasileira para as Categorias Abordadas	41
4.15	Cálculo do ICC da PT-SVM para o espaço S_1	42
4.16	Cálculo do ICC da PT-SVM para o espaço S_2	43
4.17	Cálculo do ICC da PT-SVM para o espaço S_3	43
4.18	Cálculo do ICC da PT-SVM para o espaço S_4	43
4.19	Cálculo do ICC da PT-SVM para o espaço S_5	43
4.20	PT–SVM – Taxa Média de Acerto, Desvio Padrão e ICC	44
4.21	Sumário de Resultados	45

1 INTRODUÇÃO

Atualmente o processamento de imagens é uma área de conhecimento amplamente utilizada na automação de tarefas que dependem de um ou mais observadores humanos. Estas tarefas aparecem em diversas áreas, tais como segurança – na identificação de pessoas, através da face ou impressões digitais –, medicina – na procura e classificação de ocorrências médicas, como tumores cancerígenos – e tantas outras, recentemente incluindo a área de Sistemas Inteligentes de Transporte.

Este trabalho dedica-se a essa última área, mais especificamente a utilização de ferramentas de visão computacional na resolução do problema de categorização de veículos terrestres em vias de rodagem.

1.1 MOTIVAÇÃO

Com o crescimento das cidades e, por consequência, da frota de veículos, o trânsito nos médios e grandes pólos acabou se tornando um problema. Ruas mal dimensionadas, expansões viárias que não acompanham a expansão urbana e falta de políticas públicas que realmente incentivem o uso de transportes públicos são motivos que levam os motoristas a enfrentar diariamente tráfego intenso e mal organizado, o que diminui consideravelmente a qualidade de vida e a eficiência da rede de transporte.

Para evitar ou corrigir alguns dos problemas citados, os engenheiros civis, que são responsáveis pela manutenção de trânsito, recorrem a estudos que os instrumentalizam no modelamento da malha e no planejamento de alterações e melhorias. Esses estudos são baseados principalmente na contagem de veículos [1]. Essas contagens são feitas utilizando diferentes ferramentas, com diversas metodologias. No Brasil, as mais comuns são as contagens de linha e cordão, apresentadas na Figura 1.1. A primeira consiste em traçar uma linha imaginária, dividindo a região de interesse (cidade) em duas, e contar a quantidade de veículos que passa de um lado para o outro. A contagem em cordão, por seu turno, consiste em monitorar uma região, como o centro da cidade, e contabilizar o fluxo de veículos entre determinadas entradas e saídas da região em monitoramento.

No Brasil, atualmente, a coleta utilizada nas contagens é feita de duas maneiras, manualmente, por observadores humanos, ou automaticamente por laços indutivos.

A contagem manual é feita por observadores que anotam as placas em formulários, como apresentado na figura 1.2, ou ditam para gravadores portáteis. Após esta coleta, os dados são digitados e processados. Para reduzir o custo do processo – utilizando menos observadores – os engenheiros responsáveis pelos estudos utilizam simplificações estatisticamente válidas, tais como a contagem apenas de carros de determinadas cores, ou a realização de estudos somente em determinados dias e horários. Com isto, tem-se a redução da precisão dos dados e a perda da informação de sazonalidade em prol do custo.

Com a redução considerável dos preços de câmeras digitais de vídeo e de dispositivos de armazenamento, a contagem por vídeo apresenta considerável melhora na precisão dos resultados e na redução dos custos de contagem. A contagem manual em vídeo é feita com a captura de sequências de vídeos em pontos e horários de interesse, que são posteriormente contabilizadas de forma manual por um observador humano.

Logo, torna-se necessária uma equipe pequena em campo, para executar as filmagens, e uma equipe pouco qualificada para fazer a contagem *off-line* de veículos diretamente do vídeo. Com o uso desta metodologia, aumenta-se consideravelmente a qualidade dos resultados, pois as imagens são feitas a partir de um ponto de vista privilegiado, localizado a uma altura considerável, o que diminui a probabilidade de ocorrência de oclusões. Como é possível assistir aos vídeos tantas vezes forem necessárias, a contagem pode considerar todos os veículos, sem simplificações, gerando dados mais próximos à realidade.

A primeira tentativa de automação do processo de contagem volumétrica foi com o uso de equipamentos eletrônicos baseados em laços indutivos. Porém este tipo de equipamento é caro e, na maioria das vezes, não portátil. Assim, apesar de ser capaz de executar a categorização, ele é utilizado basicamente para realizar contagem volumétrica simples, sem realizar identificação dos veículos nas entradas e saídas das regiões de interesse, informação esta requerida nos estudos de contagem em linha e cordão.

1.2 OBJETIVO E JUSTIFICATIVA

Este trabalho está inserido em um projeto que tem como objetivo principal executar contagens do tipo de linha e cordão utilizando câmeras filmadoras, sem a necessidade de observadores humanos ou laços indutivos. Além de contabilizar o fluxo de veículos, é necessário categorizá-los, uma vez que cada tipo de veículo tem suas próprias características, e a quantidade de veículos de cada categoria é um fator importante ao planejamento de tráfego, já que, devido às diferenças de tamanho, peso e características de utilização, diferentes categorias afetam o tráfego de diferentes formas.

Para este trabalho foram definidas quatro categorias de interesse para os planejadores de tráfego: carros leves, *vans*, ônibus e caminhões. Exemplos destas categorias são apresentados na figura 1.3.

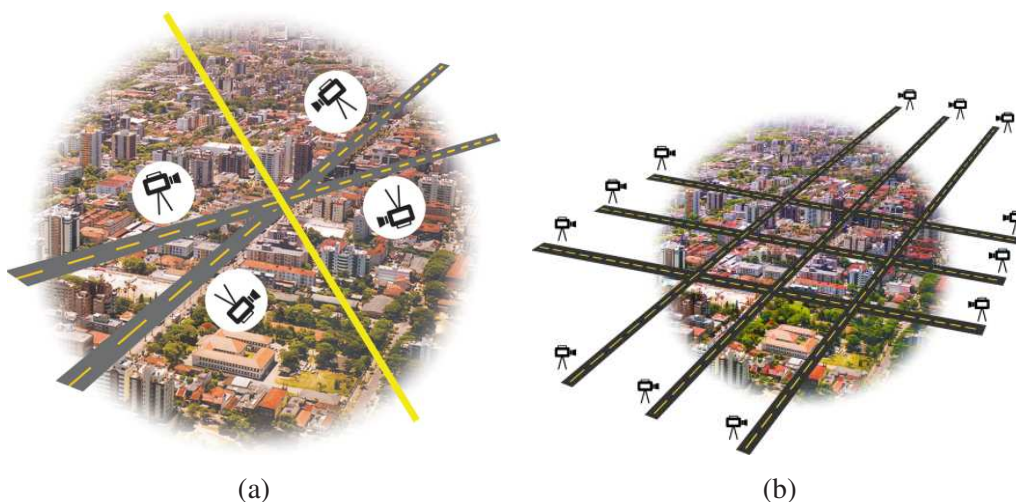


Figura 1.1: Esquemas de Contagem: a) Linha e b) Cordão



(a)



(b)



(c)



(d)

Figura 1.3: Categorias de Veículos: a) Carro leve, b) van, c) ônibus e d) caminhão

2 REVISÃO BIBLIOGRÁFICA

A categorização de veículos é um item importante nas medidas de tráfego quando se trata da realização de estudos e modelagem. Como as categorias requeridas diferem de aplicação para aplicação, o problema da categorização não tem padronização de classes, até mesmo porque cada país tem uma legislação diferente, com limites de ocupação e carga distintos.

2.1 CATEGORIZAÇÃO DE VEÍCULOS

A literatura a respeito de categorização de veículos terrestres por métodos não baseados em visão computacional se concentra em duas frentes: a de laços indutivos e a de extensômetros (*strain gauges*).

O campo mais desenvolvido consiste na utilização de laços indutivos que, apesar de terem altos custos de instalação, permitem a extração de dados por longos períodos sem a necessidade de manutenção, por serem instalados em fendas estreitas no pavimento.

Por ser um campo explorado há bastante tempo, já existem estudos comparando configurações, disposições e características físicas dos laços [4] para a categorização de veículos e sub-tarefas específicas, como a contagem de eixos.

Em [5], o modelo proposto é otimizado para a contagem de eixos. Com base em treze características da assinatura magnética dos veículos, ele é capaz de distinguir sete categorias com taxas de acerto entre 82% e 87%.

Shin *et al.* [6] propõem um sistema composto de extensômetros para a categorização de veículos nas seguintes classes: veículos pequenos, caminhões médios, ônibus e caminhões grandes, caminhões de 3 eixos e caminhões articulados. O conjunto de dados é composto por 500 amostras para treino, 100 para validação e 100 para testes, para cada uma das cinco classes. Após o pré-processamento, é usada a técnica *Principal Component Analysis* (PCA) para a extração de características, sendo escolhidas 27 para serem utilizadas em classificadores Bayesianos, RNA e SVM. As taxas de acerto relatadas foram de 86,2%, 94,2% e 94,8%, respectivamente.

2.2 CATEGORIZAÇÃO DE VEÍCULOS POR VISÃO COMPUTACIONAL

A visão computacional vem sendo utilizada na solução de problemas de tráfego por diversos pesquisadores. Estes sistemas são compostos normalmente de três etapas: detecção, rastreamento e categorização de veículos. A primeira etapa é normalmente implementada por algoritmos de subtração de fundo e filtros morfológicos. A segunda é responsável pelo rastreamento do mesmo veículo por diversos quadros, com o objetivo de extrair o caminho utilizado. Diversos métodos são utilizados para este fim, como o de filtros de Kalman. A terceira etapa é responsável por classificar um veículo em uma das categorias definidas pelos pesquisadores. É nesta última etapa que o trabalho e esta revisão bibliográfica estão focados.

O trabalho de Ji *et al.*[2] utiliza um banco de filtros de Gabor para a extração de características e a classificação de veículos entre cinco classes, apresentadas na figura 2.1.

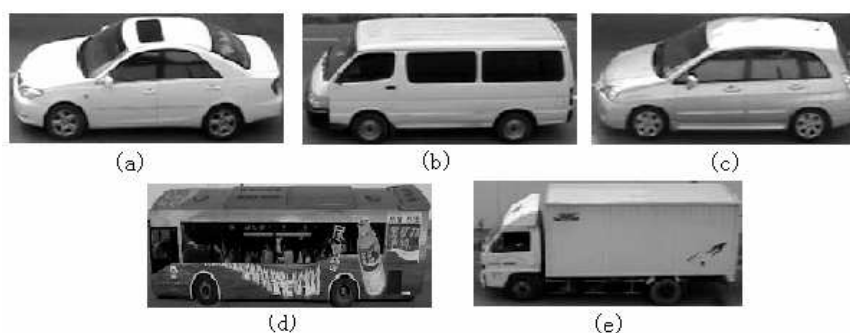


Figura 2.1: Classes utilizadas em [2]: a) carro *sedan*, b) van, c) carro *hatch*, d) ônibus e e) caminhão.

Este trabalho faz uso de uma técnica de amostragem que busca diminuir a quantidade de pontos a serem filtrados, já que a filtragem utilizando um banco de filtros de Gabor é computacionalmente cara, para diminuir a redundância de informações. Após a extração, o conjunto de características tem sua dimensão reduzida através de PCA, sendo a seguir utilizado um classificador de distância mínima.

A maior taxa de acerto reportada é de 95,17%, para um conjunto de testes de 946 veículos, utilizando um banco de 32 filtros e amostragem parcial, que considera os *pixels* da metade superior da imagem, sub-amostrados com espaçamento de 8 *pixels*. As imagens utilizadas para os trabalhos são de perfis de veículos, não tendo grande variância na posição e na distância da câmera em uma mesma categoria.

Em [3], é proposto o uso de um classificador baseado em três características extraídas da silhueta dos veículos (comprimento, proporção entre altura e largura e fator de forma (*shape factor*)) a fim de classificá-los em 7 classes, apresentadas na figura 2.2.



Figura 2.2: Classes utilizadas em [3]: a) carro *sedan*, b) van, c) camioneta, d) caminhão aberto, e) caminhão baú, f) ônibus e g) caminhão articulado.

A classificação é feita em duas etapas: a primeira separa veículos longos de curtos por meio do comprimento. Se o veículo for pequeno, a proporção e o fator de forma são somados, de forma ponderada, e o resultado é comparado a um limiar para definir se o veículo é um carro *sedan* ou uma van/camioneta. A separação entre estas últimas classes ocorre através do fator de forma. No caso de o veículo ser grande, as quatro classes restantes são separadas em ônibus/caminhão baú e caminhão aberto/articulado, utilizando apenas o fator de forma. Ônibus e caminhão baú são separados utilizando a razão entre altura e largura, e os caminhões aberto e articulado são classificados pela soma ponderada do comprimento e da razão entre comprimento e área. As imagens usadas mostram que não há variação no ponto de vista da câmera, por

esta estar fixa ao lado de uma estrada e ser posicionada de maneira perpendicular à via. A taxa de acerto apresentada neste trabalho é de 91,35% para um total de 312 veículos avaliados.

Alguns trabalhos utilizam um número menor de categorias de veículos devido à limitação de poder computacional disponível. Em [7], apenas duas classes são escolhidas: carros (a maioria dos veículos) e não-carros (vans, caminhonetes, caminhões e ônibus). As características utilizadas são comprimento e uma composição de altura e largura. Como a câmera tem posição fixa, é possível executar a calibração do sistema por meio de parâmetros intrínsecos da câmera e de dados conhecidos como posição da câmera em relação à via.

A taxa de acerto reportada é de aproximadamente 70% (a quantidade de veículos de teste não foi citada). A maioria dos erros ocorre devido à semelhança das classes nestas duas características. A classificação é feita através de uma função de discriminação com fatores obtidos por meio das médias das características para um conjunto de treino formado por 50 veículos de cada classe.

Xiong *et al.* [8] propõem o uso de transformada *wavelet* para a classificação. Inicialmente as imagens de veículos passam por uma transformação geométrica de perspectiva, para depois receberem a transformada de *wavelet*. A imagem resultante da transformada tem sua dimensão reduzida para 256 valores e classificada através de uma RNA. As classes utilizadas são: motocicletas, carros / vans / caminhonete / reboque, ônibus / caminhão de dois ou três eixos, caminhões articulados com dois a oito eixos e caminhões articulados com mais de oito eixos. O total de veículos utilizados para teste é de 341, atingindo uma taxa de acerto de 90%.

O trabalho descrito em [9] apresenta duas formas de se categorizar veículos em 7 classes, com as classes *sedan*, caminhão baú, utilitário, caminhão articulado, van, caminhão baú/utilitário/van e caminhão de mudança. A primeira consiste em extrair 17 características, 11 diretamente da região do veículo, como área, perímetro de polígono convexo, comprimento, centróide e cinco momentos, de um único quadro. Estas 17 características são transformadas, via *Linear Discriminant Analysis* (LDA), em um vetor de 5 características, classificado por um algoritmo de *k - NN* ponderado, resultando em 74,4% de acerto.

A segunda abordagem em [9] retira as mesmas 17 características do mesmo veículo em cada quadro de uma sequência de vídeo. Usando um algoritmo de máxima semelhança, a classe do veículo é atualizada a cada quadro. A classificação de várias imagens do mesmo veículo aumenta o desempenho do sistema para 88,4%.

3 METODOLOGIA UTILIZADA

Este capítulo apresenta as técnicas utilizadas neste trabalho, para a extração e a seleção de características e classificação.

3.1 EXTRAÇÃO DE CARACTERÍSTICAS

Após a seleção da região que compreende o veículo, as características são extraídas. Por este trabalho ter foco em baixa complexidade computacional, características de extração barata e eficácia conhecida foram selecionadas.

As características escolhidas para este trabalho são compostas por momentos principais e momentos invariantes à escala e à translação, calculados a partir da máscara binária que representa o espaço ocupado pelo veículo na imagem.

Momentos são médias ponderadas das intensidades dos pontos de uma imagem e que representam objetos depois da segmentação podem informar, entre outras características: centro de massa, orientação e excentricidade [10].

Estudos têm demonstrado a capacidade de momentos, ou de suas combinações não-lineares, de descrever classes de objetos [11]. Em [12], momentos foram utilizados com sucesso para classificar letras, enquanto [13] utilizou momentos para executar o reconhecimento de faces por meio de um classificador baseado em RNA.

Por se utilizar apenas a máscara binária que representa o veículo, algumas simplificações podem ser feitas na implementação. A principal delas é o cálculo somente nos pontos que a máscara não é nula, ou seja, igual a um. Esta simplificação reduz consideravelmente a complexidade computacional da extração, uma vez que as imagens que contêm a máscara são compostas por uma região nula considerável, conforme apresentado na Figura 3.1, composta pela imagem original e pela máscara obtida através de segmentação manual.



Figura 3.1: Geração de Máscara: a) Imagem Original e b) Máscara Obtida

Para que a diferença entre momentos principais e invariantes possa ser verificada de forma efetiva na

etapa de seleção de características, a imagem utilizada não sofre nenhum tipo de redimensionamento ou normalização

Técnicas de momentos são utilizadas neste trabalho por fornecerem um conjunto de características geométricas e de forma da imagem que são amplamente utilizados em problemas de reconhecimento de padrões [10]. Em estatística, momentos de ordem 0, 1 e 2 de uma dada função de densidade de probabilidade (*pdf*) representam a probabilidade total, a esperança e a variância da função, respectivamente. Em mecânica, fornecem a massa total, o centro de massa e a inércia de um certo corpo. Considerando uma imagem em escala de cinza, estes mesmos momentos fornecem a área da imagem, a sua centróide e a sua excentricidade. Enquanto momentos iniciais representam características gerais da imagem, momentos de ordem mais alta representam características mais específicas, porém, são mais sujeitos a ruído.

Com pesquisas efetuadas em análise de imagem, muitos tipos novos de momento foram introduzidos, como momentos invariantes à rotação, translação e escala. Este trabalho propõe o uso de momentos principais (também conhecidos como momentos geométricos) e momentos invariantes à translação e à escala. Momentos principais foram escolhidos pela sua baixa complexidade, enquanto os momentos invariantes pela sua robustez a mudanças do ponto de vista. Nas seções abaixo há uma breve descrição destes dois tipos de momentos e de como será efetuada sua aplicação nas imagens.

3.1.1 Momentos Principais

Os momentos principais m_{pq} podem ser calculados por

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y) \quad (3.1)$$

em que p e q definem a ordem horizontal e vertical do momento, x e y são as posições dos *pixels* da imagem I .

Dessa definição temos, para uma imagem binária, que o momento m_{00} representa a área da imagem e que a relação dos momentos $\frac{m_{10}}{m_{00}}$ e $\frac{m_{01}}{m_{00}}$ representa as coordenadas x e y da centróide do objeto.

Nota-se que estes momentos não possuem qualquer tipo de invariância e aumentam exponencialmente sua grandeza com o aumento da ordem dada por p e q .

3.1.2 Momentos Invariantes à Translação e ao Escalonamento

Os momentos invariantes à escala e translação são amplamente utilizados em situações em que há mudança do ponto de vista do observador [14][15]. Estes momentos são definidos por

$$\eta_p = \frac{\sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y)}{\sum_x \sum_y I(x, y)^{\frac{p+q+2}{2}}} \quad (3.2)$$

em que $I(x, y)$ é a intensidade da imagem no ponto (x, y) , \bar{x} e \bar{y} são as coordenadas da centróide da imagem. Neste trabalho os momentos são utilizados em imagens binárias definidas como:

$$I(x, y) = \begin{cases} 1, & \forall(x, y) \in M \\ 0, & \text{outros} \end{cases} \quad (3.3)$$

em que M é o domínio da máscara da imagem, que delimita a região de interesse, o veículo, para o cálculo dos momentos.

Logo, a implementação computacional dos momentos é simplificada consideravelmente. A aplicação de 3.3 em 3.1 e 3.2, resulta em

$$m_{pq} = \sum_x \sum_y x^p y^q, \forall(x, y) \in M \quad (3.4)$$

$$\eta_{pq} = \frac{\sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q}{\sum_x \sum_y 1}, \forall(x, y) \in M \quad (3.5)$$

Tal simplificação obtém uma significativa redução da complexidade computacional da etapa de extração de características.

Os valores obtidos em 3.4 e 3.5 são normalizados para que sejam limitados ao intervalo entre 0,1 e 0,9. A normalização de valores é uma técnica comum, que tem como objetivo diminuir a amplitude das características, permitindo assim que classificadores como Redes Neurais Artificiais (RNA), cujas saídas são limitadas entre 0 e 1, no caso de função de ativação sigmoideal, sejam utilizados de forma plena. A normalização das características, quando utilizada em conjunto com classificadores baseados em distância, como o k -NN e *Template Matching*, consideram todas características com o mesmo peso, uma vez que as distâncias mais utilizadas – Euclidiana e City Block – não levam em conta a diferença das escalas de diferentes dimensões. Caso a normalização não seja aplicada, um momento principal de ordem mais alta resultará em um valor exponencialmente maior, o que diminui a significância de momentos de ordem menor, para estes tipos de classificadores.

Os valores limite de 0,1 e 0,9 foram escolhidos para que uma margem de normalização exista, caso seja apresentado ao sistema uma nova característica com valor superior ao maior ou inferior ao menor valor dos conjuntos utilizados.

A normalização é feita de forma independente para cada característica, utilizando todos os vetores dos conjuntos de treino e teste. A implementação consiste no cálculo dos fatores de normalização A_{pq} e B_{pq} para cada ordem pq , tendo como limites inferior e superior os parâmetros MIN e MAX , definidos anteriormente como 0,1 e 0,9, respectivamente. Assim,

$$A_{pq} = \frac{MAX_{pq} - MIN_{pq}}{max(X_{pq}) - min(X_{pq})} \quad (3.6)$$

$$B_{pq} = MIN_{pq} - min(X_{pq}) \cdot A_{pq} \quad (3.7)$$

em que X é a característica a ser normalizada. Estes fatores são então aplicados a todo o conjunto de dados, individualmente para cada característica, calculando-se a característica normalizada \bar{X} como

$$\bar{X}_{pq} = X_{pq} A_{pq} + B_{pq} \quad (3.8)$$

Uma vez realizada a normalização, as características estão preparadas e condicionadas para serem utilizadas nas etapas de seleção de características e de classificação.

3.2 MEDIÇÃO DE DESEMPENHO

Raudys *et al.*[16] fazem uma breve revisão sobre os principais métodos de avaliação da performance de um classificador. Estes métodos podem ser analisados usando os seguintes fatores:

- A maneira como amostras multi-dimensionais são utilizadas para implementar e testar o classificador.
- A função de padrão de erros que apresenta a contribuição de cada observação do conjunto de teste para a estimativa da probabilidade de classificação errônea.

As três técnicas mais populares para a separação das amostras disponíveis como conjunto de treino e conjunto de teste são:

1. Resubstituição: todos os vetores são utilizados para implementar e treinar o classificador e são utilizados novamente para teste.
2. *Hold-Out*: supondo que o número de vetores (amostras) seja n , uma parte deste conjunto (conjunto de treino, com n_{treino} vetores) é utilizada para implementação do classificador enquanto as observações restantes ($n - n_{treino}$) (conjunto de teste) são utilizadas para cálculo da taxa de acerto.
3. *K-fold*: neste método, $\binom{n}{k}$ classificadores são implementados escolhendo k amostras das n para implementação do classificador, enquanto a taxa de acerto é medida utilizando-se $(n - k)$ vetores. Este processo é repetido para todas as topologias, e a taxa média de acerto é calculada. Na maioria dos casos, $k = 1$, sendo este caso específico conhecido como *Leave-one-out*.

A fim de realizar uma caracterização mais detalhada do desempenho do projeto de um classificador, a despeito da caracterização obtida com o uso das técnicas descritas anteriormente, é utilizado neste trabalho um procedimento exaustivo de projeto e teste de classificadores.

Dado um conjunto de vetores composto de n amostras do problema a ser resolvido, criam-se X conjuntos de vetores de treino e testes, alocando n_{treino} dos vetores disponíveis para o conjunto de treino e o restante $n_{teste} = (n - n_{treino})$ para o conjunto de teste. Esta separação deve ser feita de maneira aleatória e sem repetição. Deste modo, temos um total de $U = \binom{n}{n_{treino}}$ possíveis conjuntos de vetores de treino e testes diferentes.

Para cada conjunto de treino gerado, um classificador é projetado, seguindo a metodologia usual indicada para este classificador. Cabe ressaltar que toda metodologia e procedimentos estatísticos classicamente indicados para o ajuste dos parâmetros dos classificadores devem continuar sendo realizados da maneira

usual. A qualidade do classificador é então avaliada por meio da taxa média de acerto no respectivo conjunto de teste.

Porém, o projeto de classificadores baseado em aprendizagem estatística é bastante impactado pelos próprios conjuntos de treino e testes utilizados. Assim, uma análise estatística deste impacto é recomendada. O treinamento exaustivo de U classificadores, com conjuntos aleatórios, busca obter um modelamento da dependência da qualidade do classificador com os dados de treino/teste.

A avaliação de todas as U possibilidades torna esta metodologia, com um custo computacional extremamente elevado, inviável. No entanto, dado que os n vetores do conjunto de amostras já são uma amostragem de um espaço de vetores de entrada, a realização de uma amostragem nestes U conjuntos gerando u conjunto, apenas reduz a resolução da solução, sem causar impacto significativo a suas conclusões.

O modelo da qualidade da projeto de um classificador nos fornece a função massa de probabilidade da taxa de acerto. Neste trabalho, esta *pdf* é estimada baseando-se na Frequência Relativa de Ocorrência das Taxas de Acertos (FROTA) obtidas nos u classificadores treinados.

A FROTA é calculada através do histograma normalizado da taxa de acerto. Desse modo, para n_{teste} vetores de testes disponíveis, a resolução deste histograma será no máximo $100 \cdot \frac{1}{n_{teste}}$ %. A conclusão óbvia que quanto maior for o número de vetores de teste, mais preciso será este histograma, porém, dado a premissa que o número de amostras disponíveis é constante, um aumento no número de vetores de teste implica em uma redução no número de vetores usados para o treinamento do classificador. Esta redução, geralmente, diminui a qualidade do classificador projetado. Assim, temos um compromisso entre a qualidade do projeto do classificador e a qualidade da avaliação deste classificador. Indicamos, dessa forma, que a partição dos dados de amostras seja feita igualmente, isto é, $n_{teste} = n_{treino} = \frac{n}{2}$. A Figura 3.2 apresenta um exemplo da Frequência Relativa de Ocorrência da Taxa de Acerto.

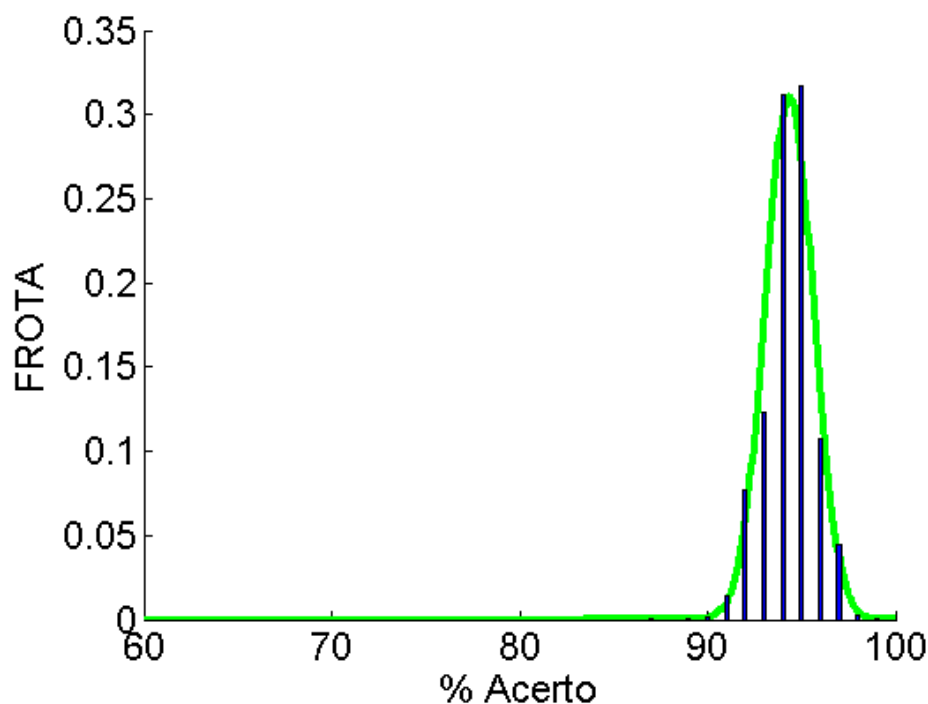


Figura 3.2: Exemplo de FROTA e Gaussiana.

Em problemas físicos naturais, como os enfrentados na área de visão computacional, a FROTA pode, em grande parte das aplicações, ser modelada por uma distribuição Gaussiana. Desse modo, o valor médio da FROTA pode ser considerado o valor esperado da taxa de acerto do classificador. Esta medida nos fornece a precisão do classificador. Assim, observa-se que classificadores mais precisos atingem taxas de acertos médias maiores que classificadores menos precisos para um determinado problema. A Figura 3.3 exemplifica estes casos.

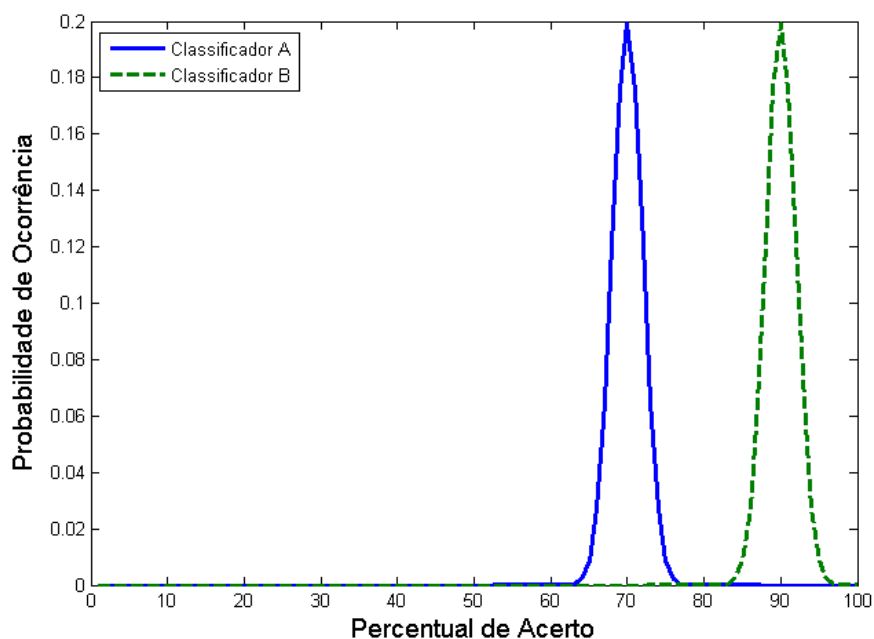


Figura 3.3: Classificadores com precisão diferentes.

O classificador A da figura 3.3 possui taxa média de acerto de 70% sendo, portanto, menos preciso que o classificador B, que atinge uma taxa média de 90%.

De modo análogo, a dispersão da FROTA ao redor da média pode ser mensurada pelo desvio padrão do modelo. Um classificador com grande dispersão possui alta sensibilidade a variações nos conjuntos de treino/teste utilizados, enquanto classificadores com pequena dispersão apresentam-se mais robustos a estas variações. Assim, a medida da dispersão da FROTA nos fornece uma idéia da robustez do classificador, isto é, o quanto a técnica de projeto do classificador (e por conseguinte o modelo empregado) é sensível às amostras de entrada. A figura 3.4 exemplifica estes casos.

O classificador B da figura 3.4 possui alta dispersão, tendo seu desvio padrão calculado em 6. Já o desvio padrão do classificador A é de 2, sendo portanto mais robusto às variações dos vetores de entrada em seu treinamento.

Com o uso desta metodologia, podemos avaliar as etapas de extração e seleção de características, e sua adequação a um determinado classificador, de modo imparcial, sem incorrer em casos especiais que possam afetar as conclusões obtidas.

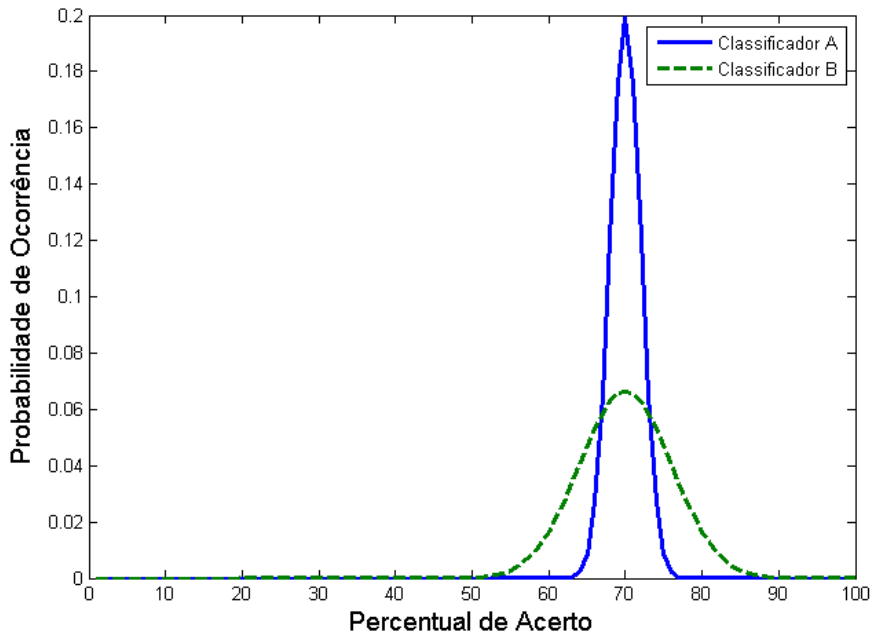


Figura 3.4: Classificadores com robustez distintas.

3.3 COMPLEXIDADE COMPUTACIONAL

Para avaliar a complexidade computacional dos métodos de extração de características e classificação foi utilizado um índice de complexidade computacional (ICC) que analisa diversas operações lógicas e matemáticas, com seu desempenho normalizado em relação ao tempo necessário para a execução de uma determinada operação.

Este estudo consiste na análise de tempo dispendido para executar determinada operação 10^{12} vezes, em máquinas de arquiteturas, hardware e software diferentes. Foram utilizados :

1. computadores com processador AMD Athlon, discos rígidos SCSI, pertencentes a um *cluster*, executando linux;
2. computadores com dois processadores Intel Pentium IV e demais características iguais a da máquina acima;
3. computadores com processador AMD Sempron e Athlon, discos IDE PATA, com sistema operacional Windows XP.

A rotina de testes foi executada em um total de 17 máquinas. Pelo grande número de operações mensuradas, alguns testes demoraram mais de dez dias para serem concluídos.

Os tempos de todas as operações de uma máquina foram então normalizados pelo tempo de execução de uma operação de soma, a fim de que os resultados obtidos pudessem ser aplicados, qualquer que fosse a máquina utilizada. A média dos resultados de todas as máquinas foi feita e o ICC calculado. A validação deste índice foi feita por meio de uma série de procedimentos[17] em que tiveram seu tempo de execução

medido em diversas máquinas. Percebeu-se que as razões entre o ICC dos diversos procedimentos eram condizentes com a razão entre seus tempos de execução.

A tabela 3.1 apresenta os resultados deste estudo, já com os pesos normalizados.

Tabela 3.1: Pesos das Operações

Operação	Peso
Soma	1
Subtração	1
Multiplicação	1
Divisão	1
Logaritmo	36.5
Exponencial	30.5
Raiz Quadrada	10.35
Potência x^y	74.35
sen	18.45
sen^{-1}	50.45
cos	14.9
cos^{-1}	49.3
tan	19.6
tan^{-1}	29.65
Acesso à Memória RAM	0.85
Escrita em Memória RAM	1.23
Comparação $i f$	0.02
Atribuição	0.02

O ICC é obtido através desta tabela, sendo calculado pelo somatório da quantidade de cada operação ponderada pelo seu peso,

$$ICC = \sum_{op} Q_{op} \cdot P_{op} \quad (3.9)$$

em que op representa a operação a ser contabilizada, Q_{op} a sua quantidade de ocorrências e P_{op} o peso dessa operação, segundo Tabela 3.1.

Cabe ressaltar que dependendo dos avanços tecnológicos na área de arquitetura de computadores, pode ser necessária a re-estimativa dos pesos da Tabela 3.1.

3.4 SELEÇÃO DE CARACTERÍSTICAS

Após a definição da metodologia para a extração das características, é importante determinar a significância de cada uma para a solução do problema proposto. A seleção de características é uma técnica que, baseada em um conjunto de D características, visa selecionar um sub-conjunto de d características, com o objetivo de otimizar algum critério, como complexidade computacional ou taxa de acerto do classificador.

Em problemas de reconhecimento e classificação de padrões, esta técnica é importante pois, em muitos problemas práticos, há um grande número de características, algumas com grande quantidade de ruído ou de baixa significância para a solução do problema. O uso destas características aumenta o custo computacional da classificação e pode reduzir a taxa de acerto do algoritmo. Em casos em que o conjunto de treino possui poucas amostras, a alta dimensionalidade do problema pode resultar em rápida especialização do classificador. Por outro lado, a escolha de poucas características pode resultar em falha na capacidade de separação das classes [18].

Para D características, é necessário avaliar $\sum_{k=1}^D \binom{D}{k}$ possibilidades de escolhas para seleção, número que pode impossibilitar computacionalmente a obtenção do sub-conjunto ótimo.

Um conjunto de técnicas de seleção de características conhecido por *Sequential Floating Search Methods (SFSM)*[19] permite a busca rápida de um conjunto sub-ótimo de d características. O *SFSM* é composto por duas técnicas básicas.

3.4.1 *SFFS – Sequential Floating Forward Selection*

A *SFFS* executa a seleção de d características a partir da ordenação de todas as D características, de forma decrescente à influência para um critério definido, como, por exemplo, o tempo gasto para a execução da tarefa, a taxa de acerto ou uma combinação dos dois. Com o constante aumento do poder computacional, a complexidade computacional vem sendo desconsiderada em prol da taxa de acerto do sistema.

Inicialmente, define-se um conjunto vazio de características escolhidas ($d = 0$). Na primeira iteração, de d iterações, os classificadores são projetados e testados com apenas uma característica. A característica que obtiver o melhor desempenho médio é fixada, ou seja, ela é considerada a melhor característica para a solução do problema, caso a classificação fosse efetuada utilizando vetores de apenas uma dimensão. Na segunda iteração, cada uma das características restantes compõem, juntamente com a característica já fixada, os conjuntos de treino e teste. O desempenho é novamente analisado e a dupla que obtiver melhor desempenho é fixada. O processo segue até que o conjunto de características escolhidas esteja completo, ou seja, o vetor de características seja formado pelas d melhores características.

3.4.2 *SFBS – Sequential Floating Backward Selection*

Enquanto a *SFFS* adiciona componentes até a quantidade de características atingir d , a *SFBS* executa um procedimento análogo, porém de forma reversa.

O conjunto de características é iniciado com todos os D componentes, e a quantidade é diminuída a

cada iteração, até que o conjunto possua d características. As iterações são compostas da retirada individual de cada uma das características ainda presentes. Para cada característica retirada, o critério – taxa de acerto – é estimado. A característica a ser excluída é aquela cuja ausência resulta em maior taxa de acerto, e assim sucessivamente para as demais iterações entre D e d .

Como a categorização de veículos é um processo composto por mais de uma etapa – extração de características e classificação, deve-se selecionar mais de um conjunto de características para executar o teste com classificadores mais robustos. Com isto, o processo de escolha do melhor conjunto de técnicas para a solução do problema proposto por este trabalho passa por duas etapas: a seleção dos melhores conjuntos de características e do melhor método de classificação.

3.5 CLASSIFICADORES CONSIDERADOS

Neste trabalho, para a tarefa de categorização de veículos baseada em momentos, será avaliado o desempenho de quatro tipos de classificadores. A validação e a escolha do melhor conjunto se dará após o uso de classificadores, como redes neurais artificiais e máquinas de vetores de suporte, além dos classificadores clássicos *k-Nearest Neighbor* e *Template Matching*.

3.5.1 k-NN

k-Nearest Neighbors [20] é um dos métodos mais simples de reconhecimento de padrões utilizados. Consiste na classificação baseada nos k exemplos de treinamento mais próximos no espaço de características utilizado.

O k define a quantidade de vizinhos a ser utilizada. Deste modo, com $k = 1$, o vetor de entrada será considerado da classe que tiver o exemplo mais próximo ao exemplo de teste. Se $k = 3$, a classe escolhida será a que tiver ao menos dois exemplos mais próximos à entrada. Para k superior, a classificação é feita por meio do esquema de votação.

Neste método, a principal definição é da a escolha da distância a ser empregada. As distâncias normalmente utilizadas são Euclidiana, definida por

$$D_e(Q, P) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (3.10)$$

em que $P = (p_1, p_2, p_3, \dots, p_n)$ e $Q = (q_1, q_2, q_3, \dots, q_n)$ são os vetores, no espaço n -dimensional, entre os quais se deseja medir a distância. A distância Manhattan, também chamada *City Block*, é definida por

$$D_m(Q, P) = \sum_{i=1}^n |p_i - q_i| \quad (3.11)$$

para os mesmos vetores $P = (p_1, p_2, p_3, \dots, p_n)$ e $Q = (q_1, q_2, q_3, \dots, q_n)$ do espaço n -dimensional.

A Figura 3.5 apresenta um exemplo que demonstra a fragilidade deste tipo de classificador no espaço de características de duas dimensões (\mathcal{R}^2) para a classificação em duas classes, X e O. O vetor de teste

é representado pelo quadrado. A linha contínua representa a superfície de decisão do classificador para $k = 1$, a tracejada para $k = 3$ e a pontilhada para $k = 5$, com o uso de distância euclidiana. Para o primeiro caso, o vetor seria classificado como X, para $k = 3$, como O e para $k = 5$, seria X novamente.

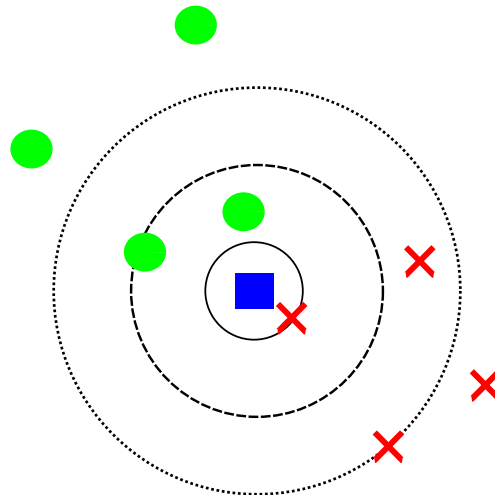


Figura 3.5: Exemplo de Classificador $k - NN$

A principal vantagem desta técnica é a dessa da execução de um algoritmo de treinamento, uma vez que a classificação consiste basicamente da comparação do vetor de entrada com os vetores que compõem o conjunto de treino. Porém, a grande quantidade de cálculos necessários para estimar a distância e escolher a classe vencedora faz com que o algoritmo seja custoso computacionalmente, tornando-o inviável em situações como a de classificação em tempo-real de vetores com grande conjunto de treino e o espaço de características de alta dimensão, sem o uso de métodos de busca rápida.

3.5.2 *TM - Template Matching*

A técnica de *TM* é similar ao $k - NN$, porém, nela, cada classe é representada por apenas um vetor, tipicamente caracterizado pela centróide da classe calculada por meio do conjunto de treinamento. Estas centróides são então utilizadas para a classificação do conjunto de testes. Como cada classe é representada por apenas um vetor, a classe vencedora consiste na que o vetor de teste esteja localizado mais próximo segundo alguma distância definida.

A complexidade computacional do *TM* é menor do que a do $k - NN$ para todos os casos em que o número de vetores de treino seja maior que o número de classes. A figura 3.6 apresenta duas classes, *O* e *X*. As centróides destas classes estão representadas pelo *O* e pelo *X* pretos. Neste caso, a classe escolhida é *O*.

Modelos (*Templates*) mais complexos são propostos na literatura, porém não fazem parte do escopo deste trabalho.

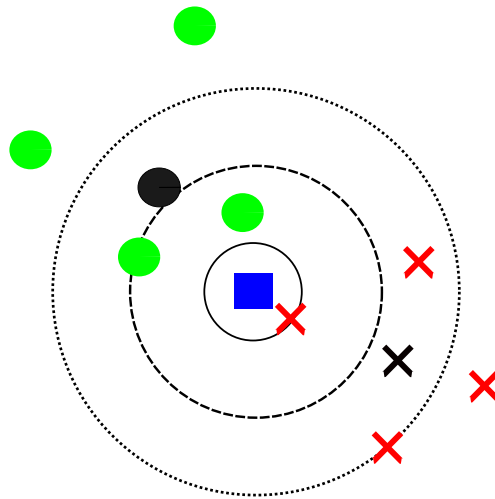


Figura 3.6: Exemplo de Classificador *TM*

3.5.3 *MLP* – Redes Neurais Artificiais *Multi-Layer Perceptron*

Redes neurais do tipo *MLP* são amplamente utilizadas para a solução de problemas de classificação por possuírem elevada capacidade de aprendizado, mesmo com um conjunto pobre de características [21]. A topologia destas redes consiste em entradas, uma ou mais camadas intermediárias e uma camada de saída, na qual acontece a apresentação do resultado. A topologia de uma *MLP* com nove entradas, uma camada escondida de seis neurônios e quatro saídas é apresentada na figura 3.7

As funções de ativação usualmente utilizadas são a tangente hiperbólica, definida por

$$\phi(v_i) = \tanh(v_i) \quad (3.12)$$

e a função logística, definida por

$$\phi(v_i) = \frac{1}{1 + e^{-v_i}} \quad (3.13)$$

em que v_i corresponde à soma das entradas e *bias* ponderadas pelos pesos. Ambas são do tipo sigmoideal, com saídas entre -1 e 1 ; e 0 e 1 , respectivamente.

O treinamento é tipicamente efetuado de maneira iterativa, apresentando-se os mesmos vetores de treinamento diversas vezes e calculando-se o erro obtido a cada iteração. Este erro é retro-propagado para as camadas anteriores em que é efetuada a atualização dos pesos. Este algoritmo de treinamento é conhecido como *BackPropagation*.

3.5.4 Máquinas de Vetores de Suporte

As máquinas de vetores de suporte são uma técnica de inteligência artificial, desenvolvida a partir de trabalhos iniciados nos anos 1960 [22], utilizadas inicialmente para classificação binária e, posteriormente, aprimoradas para problemas multi-classe [23] e de regressão, sendo esta última nomeada de *Support Vectors Regression* – *SVR*. As SVMs são capazes de aprender uma função a partir de um conjunto de

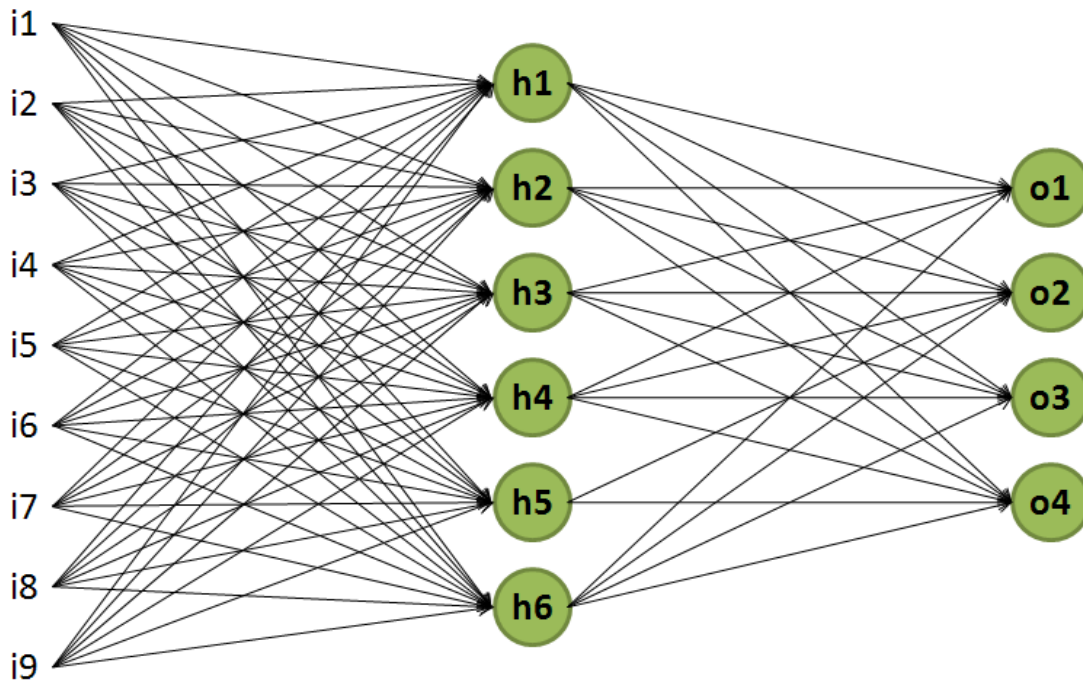


Figura 3.7: Exemplo de Rede Neural MLP

dados de treinamento, sendo consideradas como uma técnica de aprendizado supervisionado, assim como as três técnicas citadas acima.

A SVM define um hiper-plano que separa as duas classes, sendo que este hiper-plano deve maximizar a margem entre as classes. Esta margem é a distância entre outros dois hiper-plainos, um de cada lado do hiper-plano de separação. Estes hiper-plainos são calculados para ficarem o mais justo possível dos dois conjuntos de dados utilizados no treinamento. Espera-se que quanto maior a margem, melhor seja a distinção do classificador entre as duas classes.

A imagem 3.8 mostra duas classes, com dados bi-dimensionais, e três hiper-plainos possíveis: P_1 , P_2 e P_3 . Percebe-se que duas das opções (P_1 e P_2) apresentam a separação com sucesso. Porém, P_1 possui uma margem C_1 significativamente maior, sendo o melhor hiper-plano neste caso.

Originalmente, o algoritmo para cálculo de hiper-plano ótimo (com maior margem possível) era estritamente linear, tendo seu cálculo baseado em produtos internos. Posteriormente, foi feita uma adição no algoritmo, o que permitiu a criação de classificadores não lineares.

Esta adição é conhecida como *kernel trick* [24]. Ela, ao substituir o produto interno por funções não lineares, eleva a dimensionalidade dos dados que, então, podem eventualmente ser separados de forma linear nesta nova dimensão, mas de forma não linear na dimensão de origem, conforme apresentado na figura 3.9.

As funções de *kernel* $\phi(\mathbf{x}, \mathbf{x}')$ mais amplamente utilizadas são polinomial, definida por

$$\phi(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + 1)^d \quad (3.14)$$

em que d define o grau do polinômio.

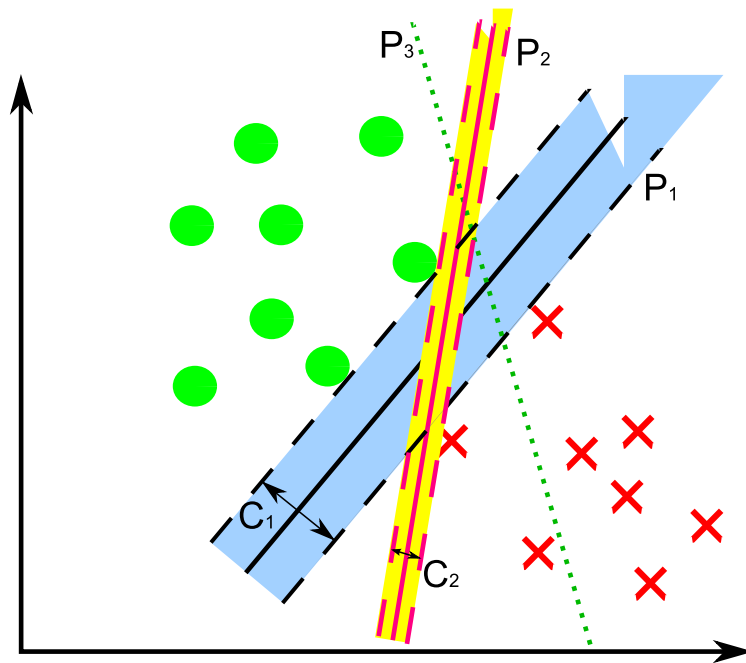


Figura 3.8: Exemplo de Hiper-planos de SVM

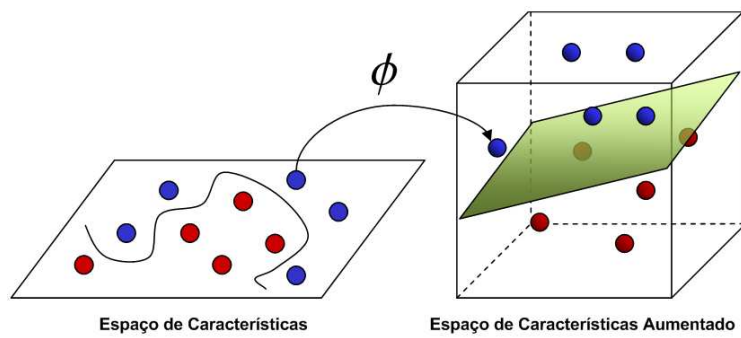


Figura 3.9: Aumento de Dimensão

RBF (*Radial Basis Function*) Multi-quadrático, definida por

$$\phi(\mathbf{x}, \mathbf{x}') = \sqrt{\|\mathbf{x} - \mathbf{x}'\|^2 + \beta^2} \quad (3.15)$$

e *kernel* Gaussiano, definido por

$$\phi(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (3.16)$$

A classificação de um vetor de entrada pela SVM é dada por meio da distância deste vetor até o hiper-plano de separação. Todavia, a magnitude da distância não é usualmente utilizada, salvo em alguns algoritmos específicos, como a classificação multi-classe. A classificação do ponto de entrada é normalmente simplificada a fim de utilizar-se somente a polaridade da distância. Ou seja, se a polaridade da distância for positiva, o ponto está em um lado do hiper-plano, sendo de uma classe, e, se for negativa, encontra-se no lado oposto, sendo membro da segunda classe.

Para a utilização de SVM em problemas multi-classe, como o apresentado neste trabalho, diversas técnicas têm sido propostas. Existem duas abordagens clássicas. A primeira utiliza a SVM para realizar a classificação entre duas classes, representada pela técnica *All versus All* (AVA). A segunda agrupa várias classes sob o mesmo *label*, possibilitando que ocorra classificações compostas por uma classe *versus* diversas classes e diversas classes *versus* diversas classes. Nesta segunda abordagem, a técnica *One versus All* é a mais conhecida.

3.5.4.1 AVA – *All versus All*

Esta técnica consiste em solucionar um problema de N classes utilizando SVMs que executem a classificação apenas entre duas classes. Para tanto, são treinadas $\frac{C}{2}$ máquinas de vetores de suporte.

A classificação é tipicamente executada por votação. Para cada SVM, a classe escolhida recebe um voto. Ao final do processo, o resultado é a classe que tiver recebido maior número de votos.

3.5.4.2 OVA – *One versus All*

Esta abordagem consiste na utilização de C máquinas de vetores de suporte, uma para cada classe, sendo a separação binária composta da classe em questão contra um conjunto composto pelas demais classes.

A classificação é feita após a execução de todas as SVMs, e a classe escolhida é a que apresentar maior distância, a seu favor, do hiper-plano de separação.

Apesar da diferença no número de SVMs necessárias para a solução do problema entre AVA e OVA, a diferença de desempenho está ligada diretamente ao número de vetores de suporte [25], e ao fato que, para classes inter-independentes, a média de vetores de suporte necessária para a separação de classes é consideravelmente superior nos esquemas OVA.

Porém, o desempenho de ambas seria próximo, pela quantidade de SVMs de cada abordagem. Segundo [26], existe pouca diferença de desempenho entre métodos mais simples, como AVA e OVA, e métodos

mais complexos. Sendo assim, é preferível utilizar os esquemas aqui apresentados, a despeito de esquemas com códigos corretores de erros [27][28].

3.5.4.3 Binary Tree SVM

O arranjo de SVMs em árvores é utilizado com o objetivo de diminuir a complexidade computacional dos testes e do uso do sistema.

Algumas topologias, como a proposta em [29], visam uma distribuição balanceada das classes, fazendo com que a maioria das classes fique no mesmo nível na árvore, com o objetivo de distribuir o custo computacional uniformemente.

Também encontram-se na literatura modelos de árvores cuja organização é efetuada pela classificação dos vetores de treino, definindo, assim, quais classes estarão posicionadas à direita ou esquerda de determinado nó [30]. A principal desvantagem deste método é a possibilidade de que uma classe seja estendida aos dois ramos de um nó, ocasionando a presença de tal classe em mais de uma folha da árvore e, provocando a redundância de SVMs necessárias para efetuar a definição de tal classe.

Existem ainda topologias que utilizam a probabilidade de ocorrência de cada classe para definir a estrutura da árvore. Um exemplo desta arquitetura é a PT-SVM [17], que tem, em sua estrutura hierárquica, cada nó contendo obrigatoriamente, no mínimo, uma classe já definida. A Figura 3.10 apresenta um exemplo de PT-SVM com a classe B sendo a mais provável, seguida por C, A e D.

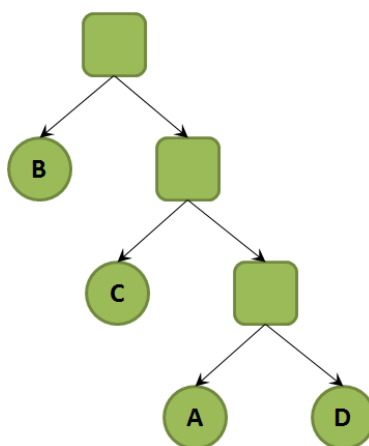


Figura 3.10: Exemplo de Classificador PT-SVM

Esta árvore é composta basicamente de várias OVA SVMs, sendo que o número de classes a cada nível é diminuído de uma classe. Esta metodologia é útil quando a probabilidade de ocorrência, *a priori* de cada uma das classes, é conhecida, e a distribuição de probabilidade não é uniforme.

4 RESULTADOS OBTIDOS

Os resultados descritos nesta seção foram obtidos através de implementações e simulações de métodos em dois tipos de ambientes. Devido ao grande esforço computacional, parte dos métodos foi implementada e executada em um cluster, utilizando a biblioteca de computação distribuída, inclusa no MATLAB.

Este cluster é formado por dois computadores com processador Intel Quad-Core e 4 GiB de memória RAM e 3 computadores com processador Intel Core 2 Duo de 1,86 GHz, com 2 GiB de RAM. Todos com um nó de processamento por núcleo, totalizando em 14 nós ligados por rede Ethernet Gigabit. Estes computadores foram responsáveis pela extração e seleção de características e pela execução dos procedimentos de treino e avaliação dos classificadores.

O segundo ambiente é formado por um computador com processador Intel Core i7, de 2,66 GHz e 8 GiB de RAM, com sistema operacional Linux. Nele foram executados os treinamentos e testes das redes neurais MLP, separados em oito processos, utilizando o software *xtlearn*.

As imagens são obtidas a partir de sinais de vídeos feitos em condições reais de uso do sistema, com um mono-pé de 5,2 metros, apresentado na figura 4.1, posicionado ao lado da via.



Figura 4.1: Mono-pé de suporte

Na parte superior deste mono-pé, uma câmera de vídeo digital é afixada. A câmera utilizada foi uma Samsung, modelo HMX-10A, que executa a captura de vídeo em uma resolução de 1280×720 pixels, a uma taxa de 60 quadros por segundo. A gravação dos vídeos é feita em uma cartão de memória SDHC, com capacidade de 16 GB, que permite a gravação de três horas de vídeo em qualidade máxima. A câmera, posicionada em seu suporte é apresentada na figura 4.2

Como o sistema deve ser capaz de executar a categorização veicular de forma robusta com relação à



Figura 4.2: Câmera e suporte utilizados

posição da câmera, imagens de filmagens distintas foram utilizadas, como pode ser visto na figura 4.3.



Figura 4.3: Imagens capturadas de diferentes pontos de vista

As imagens utilizadas foram segmentadas manualmente. A segmentação de imagens foi feita em duas etapas:

1. A seleção dos quadros que contêm veículos foi feita manualmente, utilizando um programa implementado em C++ e a biblioteca de Visão Computacional de código aberto OpenCV.
2. A segmentação dos veículos foi feita em um programa implementado em MATLAB, que permite ao usuário selecionar o veículo e a região da placa (utilizada por outra parte do sistema), e gravar os pontos que formam as seleções em um arquivo. A figura 4.4 apresenta esta ferramenta.

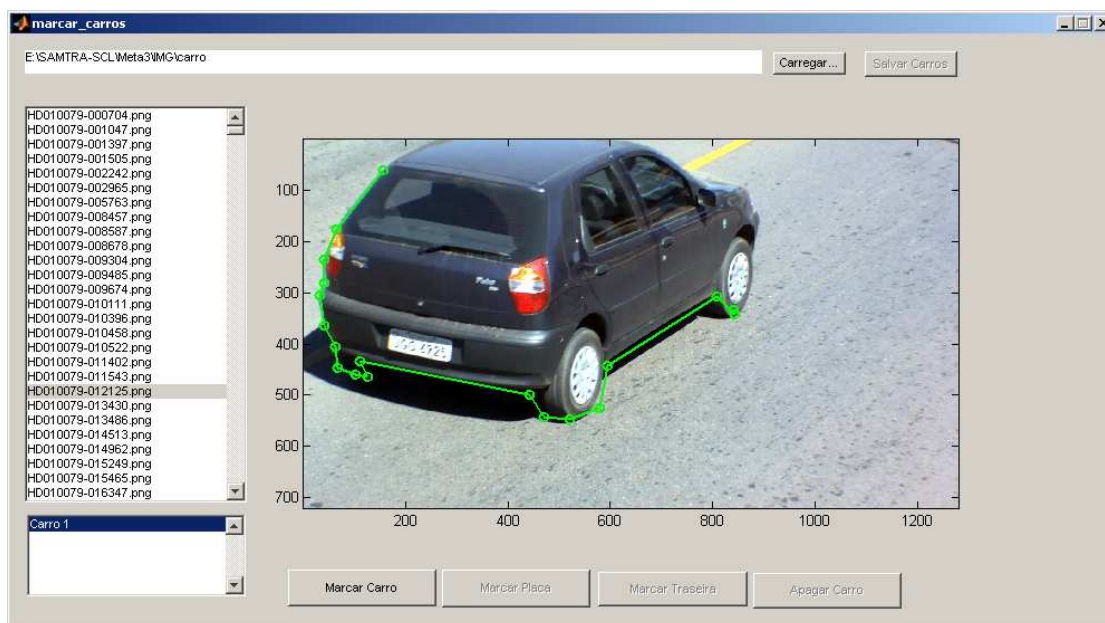


Figura 4.4: Programa para segmentação de veículos durante uso

4.1 EXTRAÇÃO DE CARACTERÍSTICAS

O grupo de características utilizado neste trabalho é composto por momentos principais e momentos invariantes à escala e translação, definidos pelas Equações 3.4 e 3.5. Foram escolhidos, para ambos os tipos de momentos, $p = 1, 2, 3$ e $q = 1, 2, 3$, totalizando nove características de momentos principais e nove de momentos invariantes.

A Tabela 4.1 apresenta exemplos de valores dos momentos principais, enquanto a Tabela 4.2 exhibe os momentos invariantes para as quatro classes de veículos consideradas e cujas imagens foram apresentadas na figura 1.3.

Tabela 4.1: Exemplo de Momentos Principais

Momento	Característica	Carro	Van	Ônibus	Caminhão
m_{11}	1	8.044591e+010	4.925273e+010	8.037840e+010	1.036713e+011
m_{12}	2	6.774659e+013	2.457158e+013	5.698745e+013	9.796108e+013
m_{13}	3	6.046791e+016	1.384654e+016	4.552419e+016	9.715871e+016
m_{21}	4	2.796292e+013	1.870999e+013	2.787920e+013	3.866932e+013
m_{22}	5	2.312875e+016	8.976779e+015	1.943571e+016	3.675008e+016
m_{23}	6	2.023057e+019	4.860999e+018	1.524645e+019	3.645401e+019
m_{31}	7	1.096719e+016	8.172626e+015	1.105445e+016	1.665622e+016
m_{32}	8	8.957429e+018	3.812890e+018	7.631094e+018	1.585961e+019
m_{33}	9	7.706218e+021	2.000752e+021	5.918976e+021	1.570505e+022

Após a extração, as características são normalizadas, segundo a equação 3.8. Um exemplo dos resultados obtidos é apresentado na tabela 4.3.

Tabela 4.2: Exemplo de Momentos Invariantes à Escala e Translação

Momento	Característica	Carro	Van	Ônibus	Caminhão
η_{11}	10	-1.401401e-002	-2.064766e-002	-3.832417e-003	1.527936e-002
η_{12}	11	5.087632e-004	-2.412944e-003	-4.101893e-003	-8.161860e-003
η_{13}	12	-4.654911e-003	-4.845673e-003	-1.898045e-003	5.945162e-003
η_{21}	13	-3.867333e-004	5.109147e-004	-9.813142e-005	8.570326e-004
η_{22}	14	4.518851e-003	6.101995e-003	6.620287e-003	7.218996e-003
η_{23}	15	-1.949883e-004	1.436451e-004	3.212581e-006	-1.227143e-004
η_{31}	16	-8.869734e-004	-2.730805e-003	-1.558427e-004	1.602762e-003
η_{32}	17	7.459970e-005	-1.101536e-004	-1.407857e-004	-3.649207e-004
η_{33}	18	-2.933078e-004	-6.113208e-004	-8.982840e-005	5.207394e-004

4.2 MEDIÇÃO DE DESEMPENHO E CONJUNTOS DE TREINO E TESTE

Para o problema abordado neste trabalho, o conjunto base é composto por 528 vetores de amostras. A separação entre conjuntos de treino e teste é feita conforme sugerido na seção 3.2, à razão 50%, formando dois conjuntos de igual tamanho para treino e teste, $n_{teste} = n_{treino} = 264$, com as classes apresentando o mesmo número de amostras nos conjuntos de treino e teste.

A quantidade de conjuntos possíveis é $U = 2.0284e + 154$, de modo que o projeto e a avaliação deste número de classificadores é inatingível em tempo razoável e com os recursos computacionais disponíveis.

Com o objetivo de reduzir o custo computacional envolvido deste universo, foram amostrados $X = 10.000$ conjuntos diferentes, que são utilizados na avaliação de todos os classificadores. Mesmo com esta amostragem, a complexidade computacional exigida é considerável, uma vez que cada classificador deve ser projetado e testado 10.000 vezes.

As tabelas a seguir apresentam a média da taxa de acerto e o desvio padrão para estes 10.000 conjuntos. Em cada tabela, o conjunto de características com melhor resultado é sombreado.

4.3 SELEÇÃO DE CARACTERÍSTICAS

Os resultados apresentados nesta seção seguem a metodologia de *SFFS*, utilizando um conjunto de tamanho $D = d = 18$, correspondente aos momentos principais e momentos invariantes à escala e translação. Com este conjunto de dezoito características, o algoritmo de seleção necessita apenas de 18 iterações para ordenar as D características pela significância na solução do problema.

Neste trabalho, assumimos que o desempenho de um classificador seja independente do espaço de características utilizado, a fim de reduzir a complexidade computacional da etapa de seleção de características. Deste modo utiliza-se o classificador $k - NN$ para estimar o conjunto de momentos mais adequado à solução do problema da classificação de veículos.

Tabela 4.3: Exemplo de Características Normalizadas

Momento	Característica	Carro	Van	Ônibus	Caminhão
\overline{m}_{11}	1	0.50562	0.61892	0.45535	0.90000
\overline{m}_{12}	2	0.35931	0.50134	0.41036	0.90000
\overline{m}_{13}	3	0.27863	0.37124	0.37914	0.90000
\overline{m}_{21}	4	0.43198	0.57083	0.37542	0.90000
\overline{m}_{22}	5	0.30518	0.45784	0.33668	0.90000
\overline{m}_{23}	6	0.23687	0.33674	0.31083	0.90000
\overline{m}_{31}	7	0.37598	0.52275	0.31128	0.90000
\overline{m}_{32}	8	0.26604	0.42926	0.27939	0.90000
\overline{m}_{33}	9	0.20731	0.31414	0.25897	0.90000
$\overline{\eta}_{11}$	10	0.42589	0.69797	0.26797	0.34622
$\overline{\eta}_{12}$	11	0.47597	0.35520	0.48642	0.41767
$\overline{\eta}_{13}$	12	0.46917	0.67163	0.39666	0.36954
$\overline{\eta}_{21}$	13	0.28074	0.21547	0.22373	0.84958
$\overline{\eta}_{22}$	14	0.18905	0.60025	0.70305	0.11314
$\overline{\eta}_{23}$	15	0.39065	0.42695	0.38273	0.65592
$\overline{\eta}_{31}$	16	0.39835	0.74719	0.24664	0.34004
$\overline{\eta}_{32}$	17	0.32907	0.23304	0.58054	0.28807
$\overline{\eta}_{33}$	18	0.47011	0.73393	0.28513	0.35564

O k-NN é um classificador extremamente simples e de obtenção de resultados rápida, além de não necessitar de projeto de parâmetros complexos, como os vetores de suporte das SVM e os pesos das MLPs.

Para aumentar a confiabilidade do resultado da etapa de seleção de características, é utilizada a metodologia de avaliação apresentada na seção 3.2. Assim, a cada iteração do algoritmo *SFFS* serão executados 10.000 testes em conjuntos diferentes. A pontuação da qualidade das características será realizada pela média dos acertos dos classificadores nos 10.000 conjuntos disponíveis, já que deseja-se prezar pela precisão do modelo.

A tabela 4.4 apresenta a média da taxa de acerto e desvio padrão, bem como a característica selecionada em cada iteração do algoritmo *SFFS*.

Pode-se observar pela tabela 4.4 que o momento $\overline{\eta}_{11}$ é a característica mais significativa, segundo o critério adotado, seguido dos momentos $\overline{\eta}_{21}$ e $\overline{\eta}_{23}$. Selecionando-se as primeiras 8 características, temos o melhor desempenho obtido do classificador, a partir do qual a incorporação de mais dimensões no vetor de entrada resulta em redução da precisão do classificador.

O gráfico 4.5 apresenta a relação entre o número de características e a *FROTA*. A cor azul representa menor frequência, enquanto a vermelha representa maior frequência relativa de ocorrência da taxa de acerto.

Com os dados da seleção obtidos, pode-se perceber que após uma elevação considerável da taxa de acerto com a seleção entre seis e oito características, há também uma diminuição na qualidade dos

Tabela 4.4: Resultado da Ordenação de Características utilizando *SFFS*

Iteração	Característica Escolhida	Momento	Taxa Média de Acerto	Desvio Padrão
1	10	$\overline{\eta}_{11}$	45.75	2.49
2	13	$\overline{\eta}_{21}$	67.09	2.27
3	15	$\overline{\eta}_{23}$	83.11	1.80
4	11	$\overline{\eta}_{12}$	90.31	1.56
5	14	$\overline{\eta}_{22}$	93.15	1.41
6	17	$\overline{\eta}_{32}$	94.42	1.31
7	18	$\overline{\eta}_{33}$	94.35	1.28
8	12	$\overline{\eta}_{13}$	94.58	1.28
9	16	$\overline{\eta}_{31}$	94.30	1.27
10	2	\overline{m}_{12}	93.81	1.30
11	3	\overline{m}_{13}	93.25	1.35
12	9	\overline{m}_{33}	92.91	1.40
13	8	\overline{m}_{32}	92.74	1.40
14	6	\overline{m}_{23}	92.47	1.44
15	7	\overline{m}_{31}	91.92	1.50
16	4	\overline{m}_{21}	91.73	1.51
17	1	\overline{m}_{11}	91.29	1.55
18	5	\overline{m}_{22}	91.09	1.56

resultados obtidos com conjuntos entre dez e dezoito características.

A partir destas análises para seleção de um conjunto sub-ótimo de características, pode-se focar o treinamento de classificadores mais elaborados e complexos, tais como MLP e SVM nos cinco espaços de características que apresentam melhores resultados, com o objetivo de verificar a veracidade da consideração da independência do conjunto de características do classificador. Estes espaços são:

$$S_1 = \{\overline{\eta}_{11}, \overline{\eta}_{21}, \overline{\eta}_{23}, \overline{\eta}_{12}, \overline{\eta}_{22}, \overline{\eta}_{32}\} \quad (4.1)$$

com 6 momentos,

$$S_2 = \{\overline{\eta}_{11}, \overline{\eta}_{21}, \overline{\eta}_{23}, \overline{\eta}_{12}, \overline{\eta}_{22}, \overline{\eta}_{32}, \overline{\eta}_{33}\} \quad (4.2)$$

com 7 momentos,

$$S_3 = \{\overline{\eta}_{11}, \overline{\eta}_{21}, \overline{\eta}_{23}, \overline{\eta}_{12}, \overline{\eta}_{22}, \overline{\eta}_{32}, \overline{\eta}_{33}, \overline{\eta}_{13}\} \quad (4.3)$$

com 8 momentos,

$$S_4 = \{\overline{\eta}_{11}, \overline{\eta}_{21}, \overline{\eta}_{23}, \overline{\eta}_{12}, \overline{\eta}_{22}, \overline{\eta}_{32}, \overline{\eta}_{33}, \overline{\eta}_{13}, \overline{\eta}_{31}\} \quad (4.4)$$

com 9 momentos e

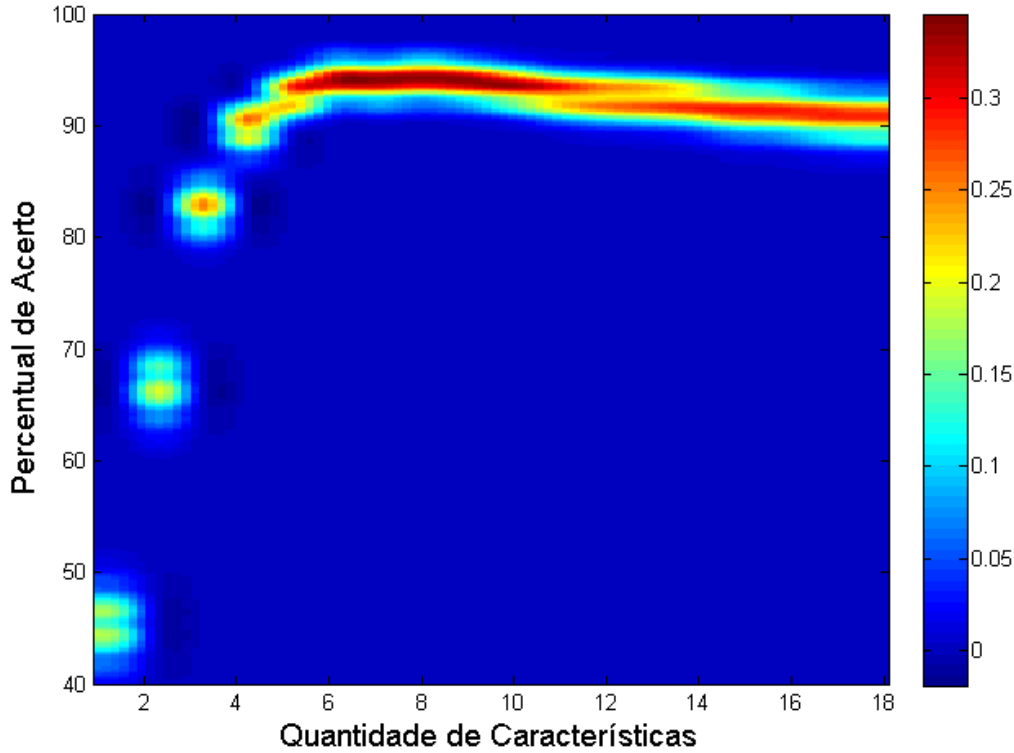


Figura 4.5: Funções de densidade de probabilidade de acerto utilizadas na seleção de características

$$S_5 = \{\overline{\eta}_{11}, \overline{\eta}_{21}, \overline{\eta}_{23}, \overline{\eta}_{12}, \overline{\eta}_{22}, \overline{\eta}_{32}, \overline{\eta}_{33}, \overline{\eta}_{13}, \overline{\eta}_{31}, \overline{m}_{12}\} \quad (4.5)$$

com 10 momentos. É possível observar que o único momento principal utilizado é \overline{m}_{12} , no espaço S_5 , depois que todos os momentos invariantes a escala e translação já haviam sido selecionados.

4.4 K-NEAREST NEIGHBOUR

Os resultados para classificação utilizando k -NN, com distância euclidiana e $k = 1$, podem ser obtidos diretamente, através do método de seleção de características citado anteriormente. Para obtenção dos resultados utilizando distância *city block*, os vetores de teste foram apresentados novamente para classificação.

4.4.1 $k = 1$

As taxas médias de acerto e o desvio padrão dos classificadores para os espaços S_1, S_2, S_3, S_4 e S_5 podem ser observados na tabela 4.5. Os mesmos itens para distância *city block* são apresentados na tabela 4.6

As FROTAs e as respectivas distribuições normais dos classificadores com distância euclidiana e *city block* são apresentadas nas figuras 4.6 e 4.7, respectivamente.

Tabela 4.5: k -NN, euclidiana, $k = 1$ – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	94,42 %	1,31	4.498
S_2	94,35 %	1,28	5.290
S_3	94,58 %	1,28	6.082
S_4	94,30 %	1,27	6.874
S_5	93.81 %	1,30	7.666

Tabela 4.6: k -NN, city block, $k = 1$ – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	93,52%	1,34	3.231
S_2	93,36%	1,31	3.769
S_3	96,70%	1,34	4.308
S_4	93,33%	1,36	4.847
S_5	93,13%	1,35	5.385

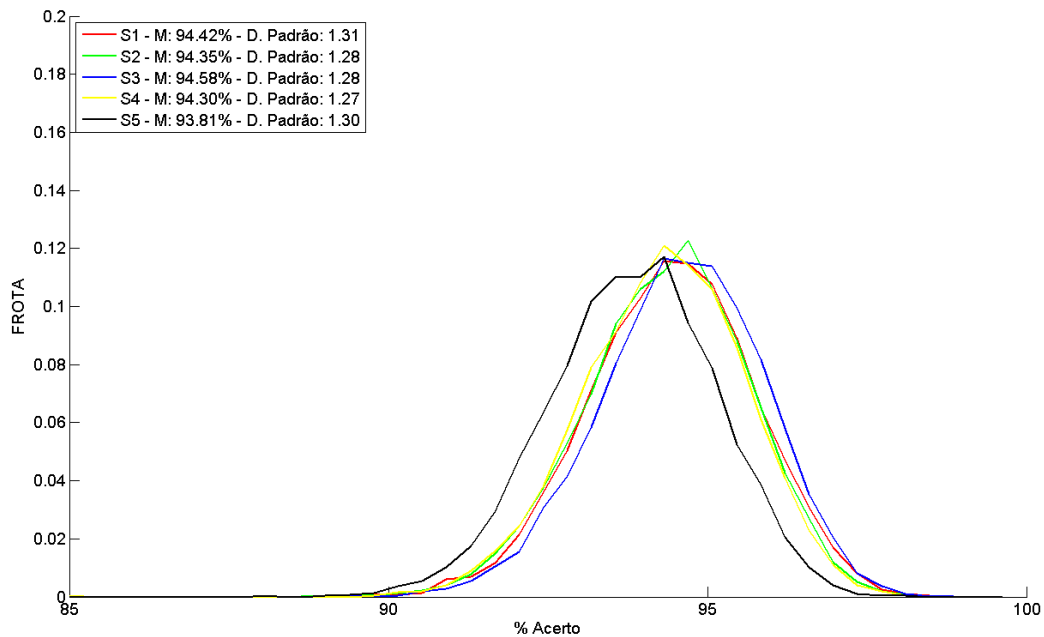


Figura 4.6: k -NN, euclidiana, $k = 1$ – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

4.4.2 $k = 3$

As taxas médias de acerto e o desvio padrão dos classificadores para os espaços S_1 , S_2 , S_3 , S_4 e S_5 podem ser observados na tabela 4.7. Os mesmos itens para distância *city block* são apresentados na tabela

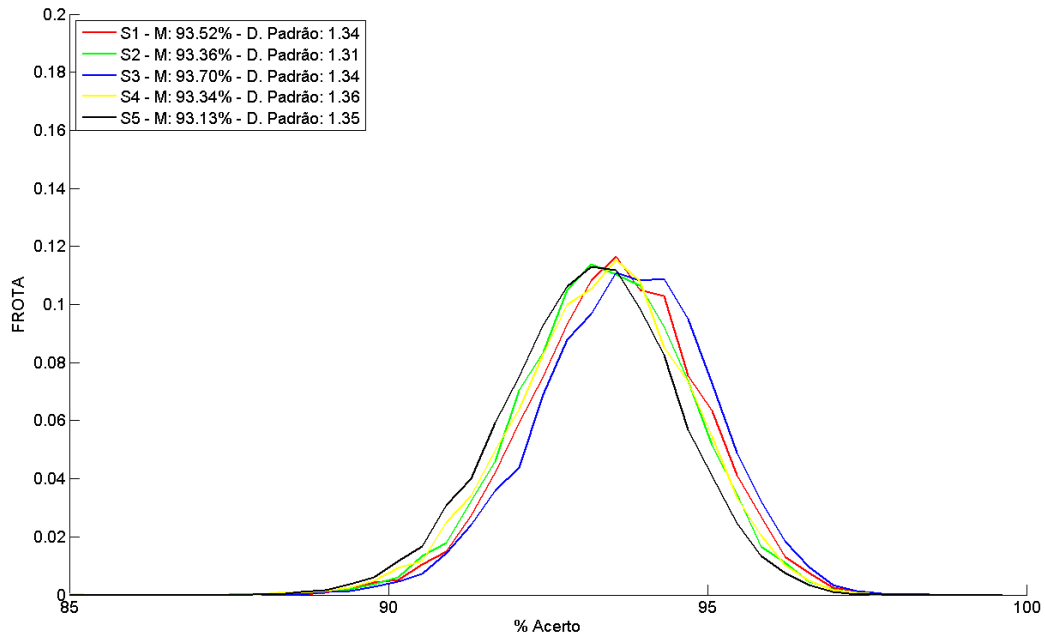


Figura 4.7: k -NN, *city block*, $k = 1$ – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

4.8.

Tabela 4.7: k -NN, euclidiana, $k = 3$ – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	93,69%	1,35	5.131
S_2	93,89%	1,28	5.923
S_3	93,94%	1,32	6.715
S_4	93,70%	1,30	7.507
S_5	92,85%	1,39	8.300

As FROTAS e as respectivas distribuições normais dos classificadores com distância euclidiana e *city block* são apresentadas nas figuras 4.8 e 4.9, respectivamente.

4.5 TEMPLATE MATCHING

Após calculadas as centróides para cada classe, por meio do o conjunto de treino, a classificação do conjunto de teste, com distância euclidiana e *city block*, foi executada para os 10.000 conjuntos. As tabelas 4.9 e 4.10 apresentam os resultados para os dois tipos de distâncias.

Tabela 4.8: k -NN, city block, $k = 3$ – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	92,99	1,41	3864
S_2	93,01	1,37	4.402
S_3	93,22	1,35	4.941
S_4	92,96	1,40	5.480
S_5	92,59	1,44	6.018

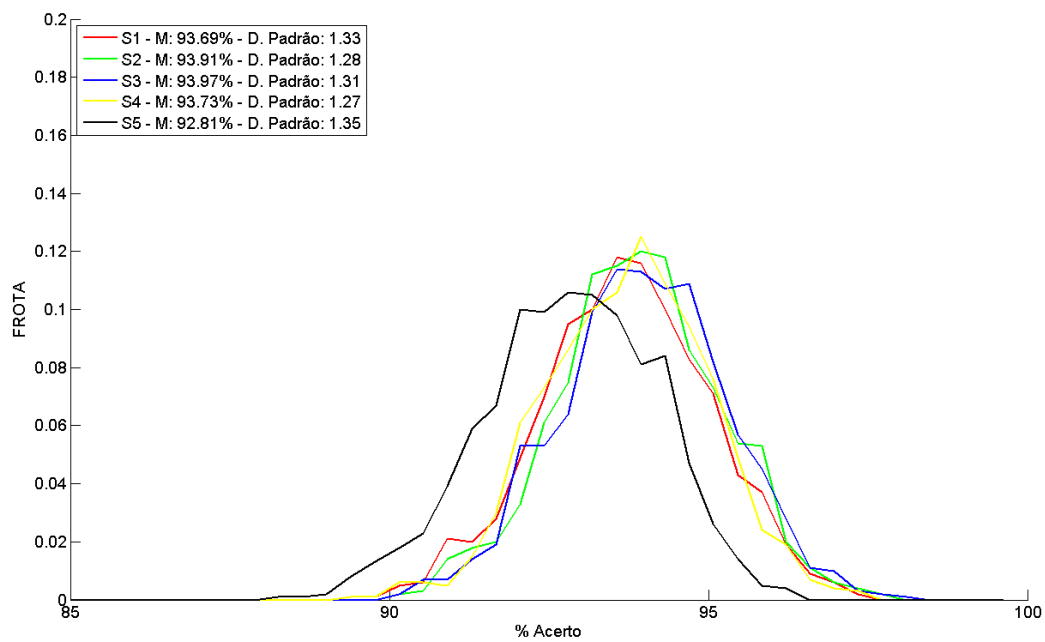


Figura 4.8: k -NN, euclidiana, $k = 3$ – FROTA nos espaços S_1, S_2, S_3, S_4 e S_5

Tabela 4.9: TM , euclidiana – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	79,53	2,04	78
S_2	78,52	2,02	90
S_3	75,84	2,16	102
S_4	77,31	2,31	114
S_5	77,84	2,36	126

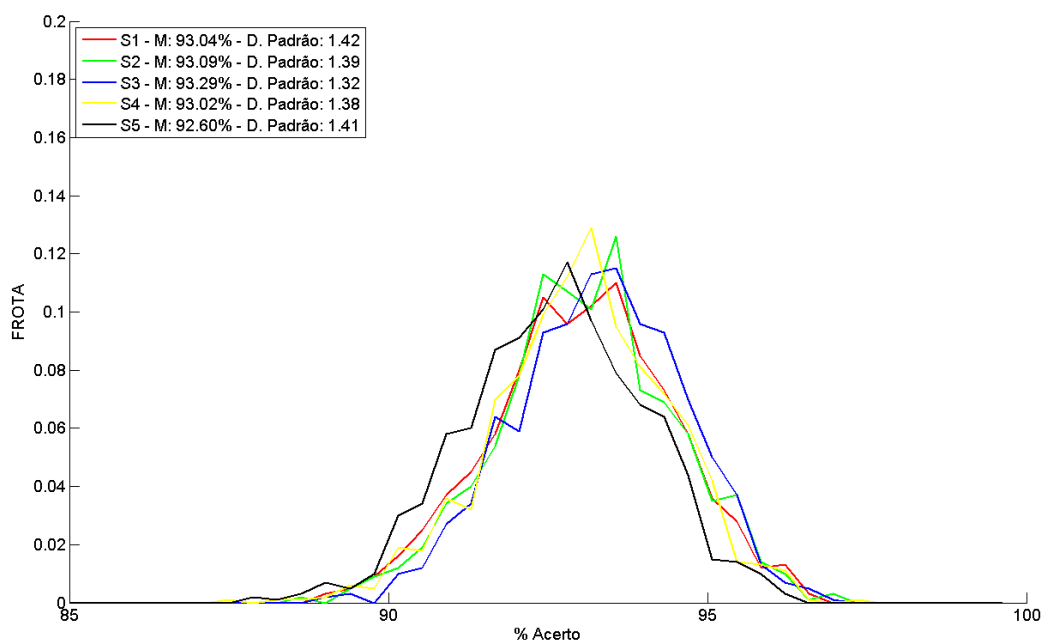


Figura 4.9: k -NN, *city block*, $k = 3$ – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

Tabela 4.10: TM , *city block* – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	77,36	2,35	49
S_2	74,09	2,41	57
S_3	69,42	2,38	65
S_4	70,06	2,73	73
S_5	70,69	2,48	82

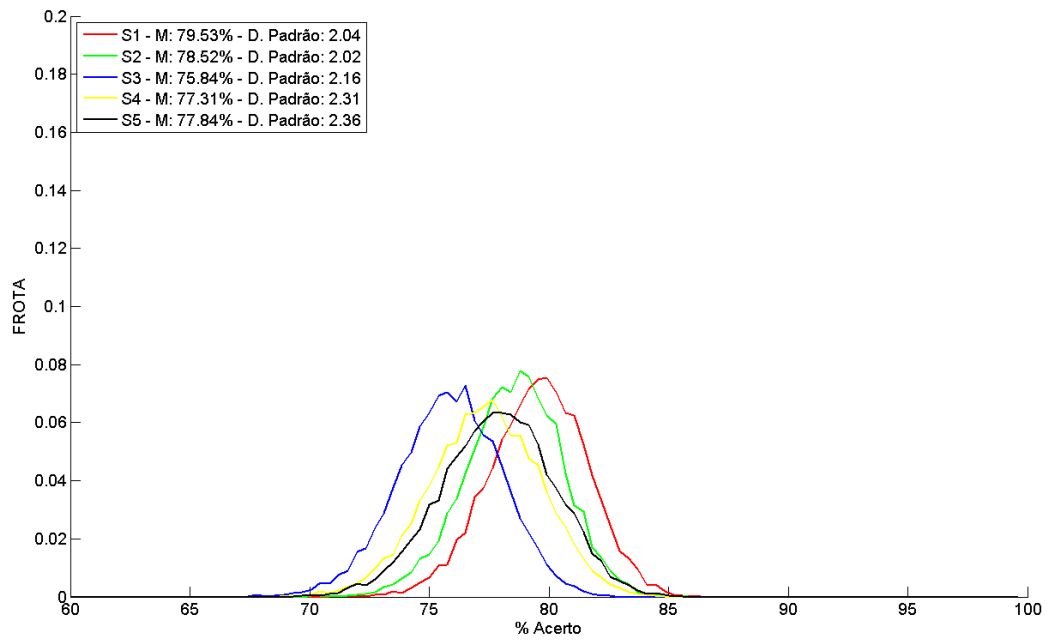


Figura 4.10: TM , euclidiana – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

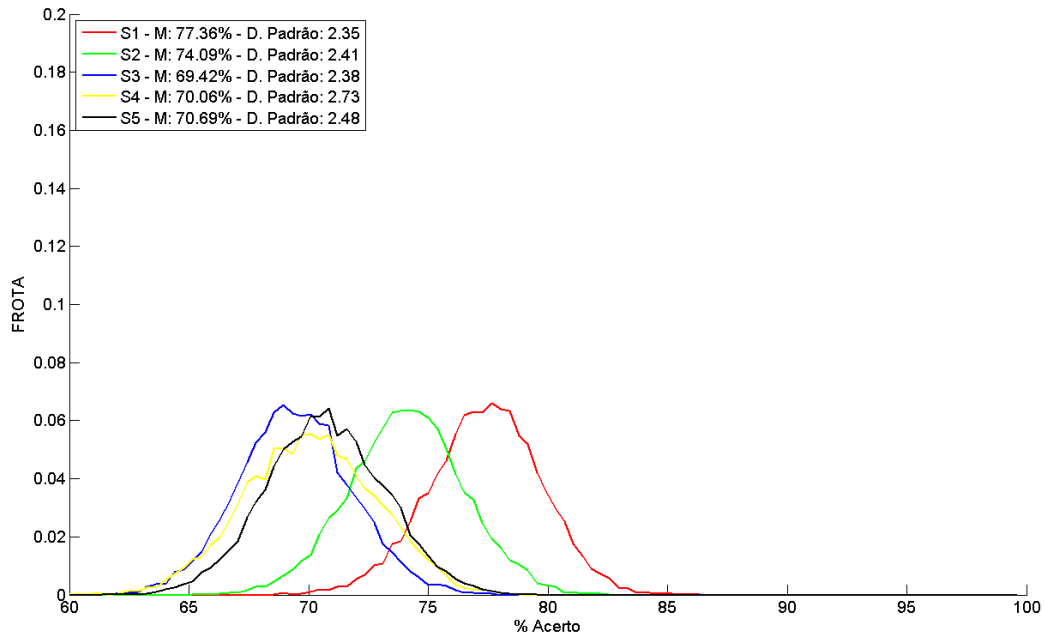


Figura 4.11: *TM, city block* – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

4.6 MLP

A definição da quantidade de neurônios na camada escondida h é dada por

$$h = \sqrt{i \cdot o} \quad (4.6)$$

em que i é a quantidade de unidades da camada de entrada e o a quantidade de neurônios na camada de saída. Para as redes dos espaços S_1 e S_2 , foram utilizados cinco neurônios na camada intermediária, enquanto para os espaços S_3 , S_4 e S_5 , seis neurônios foram alocados na camada intermediária.

Todas as redes foram treinadas com algoritmo de *BackPropagation* até 3800 épocas e taxa de aprendizado de 0.1. A raiz de geração dos números aleatórios para a inicialização dos pesos foi mantida constante para fins de repetibilidade do experimento.

A figura 4.12 apresenta as *FROTAs* individualmente, para cada espaço.

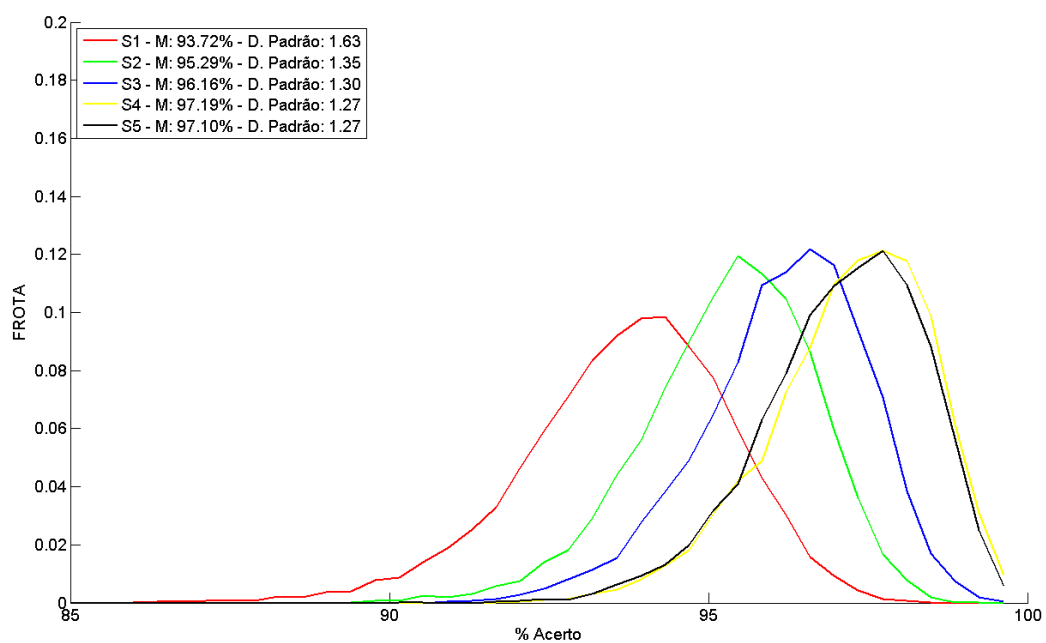


Figura 4.12: MLP – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

As taxas médias de acerto e o desvio padrão de cada uma das 10.000 implementações de MLPs projetadas é apresentada na tabela 4.11.

Tabela 4.11: MLP – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	93,72%	1,63	392
S_2	95,30%	1,35	402
S_3	96,16%	1,30	469
S_4	97,20%	1,27	481
S_5	97,10%	1,27	493

4.7 SVM – SUPPORT VECTOR MACHINES

Para que seja efetuada uma escolha melhor de classificador, além de testar cinco espaços de características, foram implementadas três topologias de SVM multi-classe: AVA, OVA e PT-SVM.

Por rigor científico, as máquinas de vetores de suporte foram treinadas com os mesmos parâmetros (*Soft Margin*, $C = 1$), e todas utilizando *kernel* polinomial de grau três.

4.7.1 AVA – All versus All

Para este problema, que consiste em quatro classes, são necessárias seis SVMs. A tabela 4.12 apresenta as taxas médias de acerto e desvio padrão da classificação dos cinco espaços de características.

Tabela 4.12: AVA-SVM – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	93,98%	1,59	649
S_2	93,70%	1,57	765
S_3	94,21%	1,49	837
S_4	93,90%	1,55	977
S_5	93,41%	1,64	1.057

A figura 4.13 apresenta o desempenho detalhado do classificador para os cinco espaços de características.

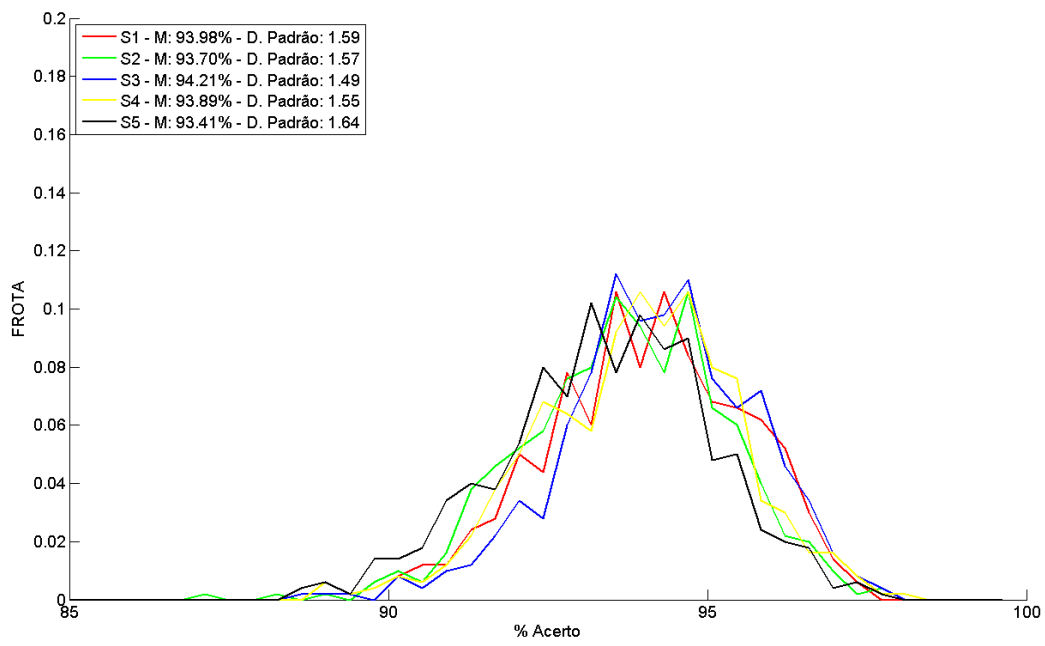


Figura 4.13: AVA – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

4.7.2 OVA – One versus All

Para este classificador é necessário o projeto de 4 SVMs. A tabela 4.13 apresenta a taxa média de acerto e o desvio padrão obtidos. O desempenho deste classificador é apresentado de forma mais detalhada na figura 4.14.

Tabela 4.13: OVA–SVM – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
S_1	92,80%	1,72	726
S_2	95,08%	1,78	804
S_3	96,59%	1,76	882
S_4	96,97%	1,78	960
S_5	96,21%	1,77	1.038

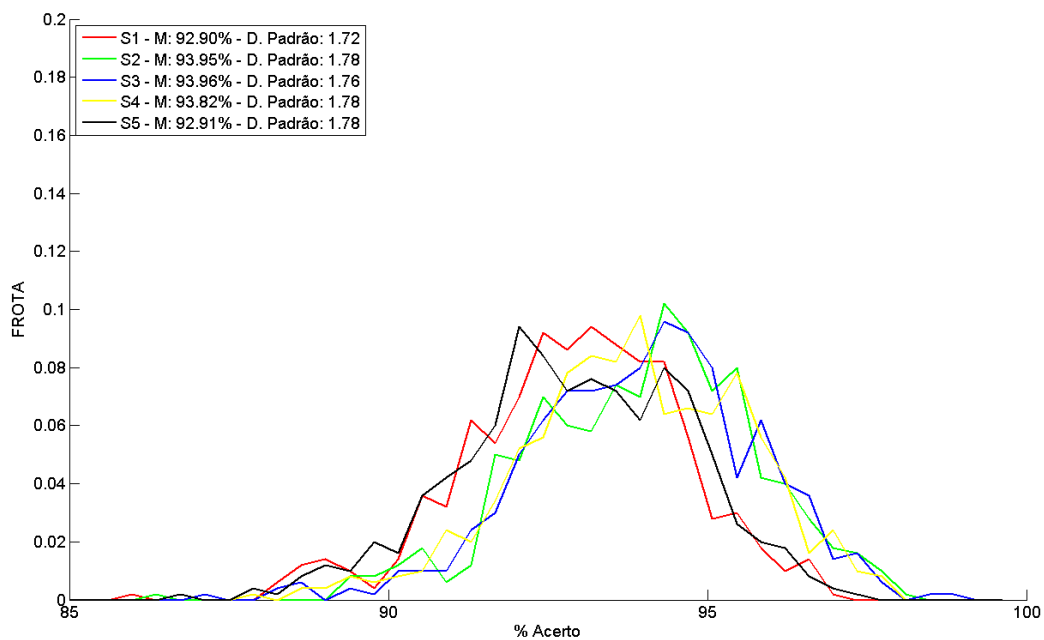


Figura 4.14: OVA – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

4.7.3 PT-SVM

Para esta topologia estruturada em árvore, a informação de probabilidade de classes é relevante, pois é responsável pela ordenação das classes na árvore.

As probabilidades utilizadas neste trabalho são baseadas em dados do DENATRAN – Departamento Nacional de Trânsito – referentes ao mês de maio de 2009 [31].

Existem 55.937.035 veículos registrado no país. Destes, deve-se selecionar as categorias relevantes, excluindo, por exemplo, motocicletas, reboques e tratores.

A tabela 4.14 apresenta os números absolutos e o percentual representado por cada categoria.

Tabela 4.14: Tamanho da Frota Brasileira para as Categorias Abordadas

Classe	Frota	Percentual
Carros	36.548.385	89,65%
Vans	1.610.884	3,95%
Ônibus	645.501	1,60%
Caminhões	1.963.100	4,80%
TOTAL	40.767.870	100%

Os percentuais de participação na frota brasileira são utilizados como probabilidade *a priori* para se gerar a PT-SVM. Logo, como os carros são os veículos com maior probabilidade de ocorrência, ficam no

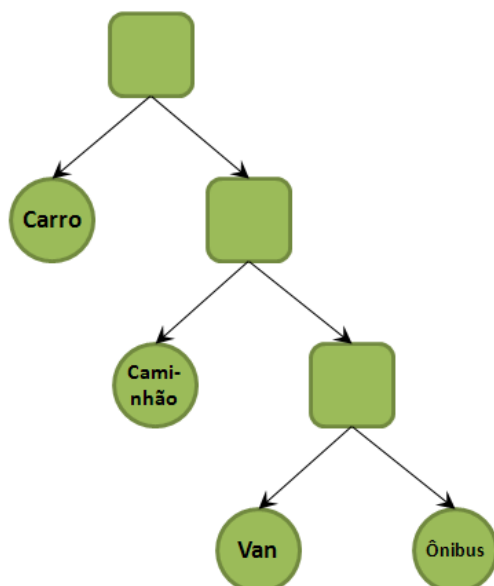


Figura 4.15: Topologia da Árvore PT-SVM

primeiro nível da árvore, os caminhões no segundo, vans e ônibus no terceiro. A topologia utilizada é apresentada na figura 4.15.

Os resultados para os espaços de características são mostrados na tabela 4.20. O cálculo do ICC para a PT-SVM leva em consideração a probabilidade de ocorrência de cada uma das classes. Sendo que, para as duas classes menos frequentes, todas as SVMs da árvore devem ser calculadas. As tabelas 4.15, 4.16, 4.17, 4.18 e 4.19 apresentam os valores da SVM para cada classe.

Tabela 4.15: Cálculo do ICC da PT-SVM para o espaço S_1

Classe	ICC da SVM	ICC Acumulado	Probabilidade	ICC Ponderado
Carros	261	261	89,65%	234
Caminhões	217	478	4,80%	23
Vans	173	912	3,95%	36
Ônibus	-	912	1,60%	15
TOTAL:				308

Tabela 4.16: Cálculo do ICC da PT-SVM para o espaço S_2

Classe	ICC da SVM	ICC Acumulado	Probabilidade	ICC Ponderado
Carros	295	295	89,65%	264
Caminhões	243	538	4,80%	26
Vans	191	729	3,95%	29
Ônibus	-	729	1,60%	12
TOTAL:				331

Tabela 4.17: Cálculo do ICC da PT-SVM para o espaço S_3

Classe	ICC da SVM	ICC Acumulado	Probabilidade	ICC Ponderado
Carros	397	397	89,65%	356
Caminhões	269	666	4,80%	32
Vans	209	868	3,95%	35
Ônibus	-	868	1,60%	14
TOTAL:				437

Tabela 4.18: Cálculo do ICC da PT-SVM para o espaço S_4

Classe	ICC da SVM	ICC Acumulado	Probabilidade	ICC Ponderado
Carros	329	329	89,65%	295
Caminhões	312	641	4,80%	31
Vans	227	868	3,95%	34
Ônibus	-	868	1,60%	14
TOTAL:				374

Tabela 4.19: Cálculo do ICC da PT-SVM para o espaço S_5

Classe	ICC da SVM	ICC Acumulado	Probabilidade	ICC Ponderado
Carros	572	572	89,65%	513
Caminhões	340	912	4,80%	44
Vans	297	1209	3,95%	48
Ônibus	-	1209	1,60%	19
TOTAL:				624

Tabela 4.20: PT-SVM – Taxa Média de Acerto, Desvio Padrão e ICC

Espaço	Taxa Média de Acerto	Desvio Padrão	ICC Ponderado
S_1	93,18%	1,63	308
S_2	94,17%	1,63	331
S_3	94,17%	1,72	437
S_4	94,14%	1,70	374
S_5	93,30%	1,69	624

As FROTAs para o classificador PT-SVM utilizando os cinco espaços de características são apresentadas na figura 4.16.

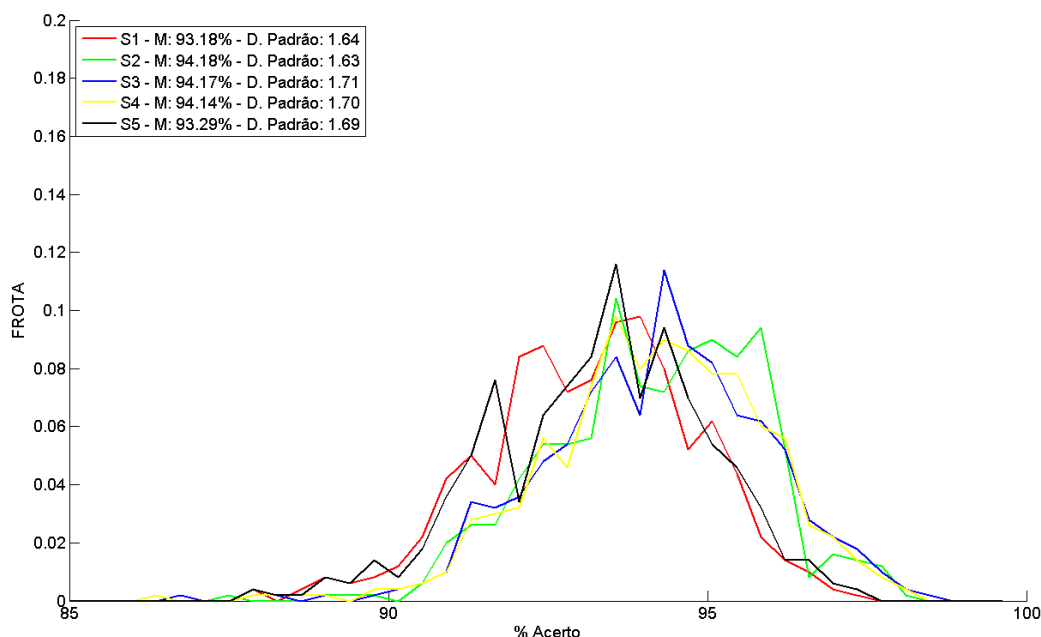


Figura 4.16: PT-SVM – FROTA nos espaços S_1 , S_2 , S_3 , S_4 e S_5

A tabela 4.21 apresenta os melhores resultados para cada método.

Tabela 4.21: Sumário de Resultados

Técnica	Espaço	Taxa Média de Acerto	Desvio Padrão	ICC
k -NN, euclidiana, $k = 1$	S_3	94,58%	1,28	6.082
k -NN, city block, $k = 1$	S_3	93,70%	1,34	4.308
k -NN, euclidiana, $k = 3$	S_3	93,94%	1,32	6.715
k -NN, city block, $k = 3$	S_3	93,22%	1,35	4.941
TM , euclidiana	S_1	79,53%	2,04	78
TM , city block	S_1	77,36%	2,35	49
MLP	S_4	97,20%	1,27	481
AVA	S_3	94,21%	1,49	837
OVA	S_4	96,97%	1,78	960
PT-SVM	S_2	94,17%	1,63	331

A técnica mais precisa e mais robusta para solucionar este problema de categorização visual de veículos baseada em momentos foi a MLP, com taxa média de acerto de 97,20% e desvio padrão de 1,27. Conforme esperado, TM é a técnica com menor custo computacional. Porém, por ela executar a classificação baseando-se apenas na centróide das classes, seu desempenho é o pior de todas as técnicas avaliadas, tanto em relação à precisão como à robustez.

O k - NN tem seus melhores resultados em S_3 , conforme esperado, devido à seleção das características ter sido executada com ele, e o espaço S_3 , composto de 8 características, ter obtido o melhor desempenho ainda na fase de seleção. A tabela 4.21 mostra que os melhores resultados desta técnica foram obtidos com distância *city block*. Porém, não há uma variância considerável nas taxas de acerto para os cinco espaços, podendo-se concluir que, para estes espaços, o k - NN não é sensível à quantidade de características. Isto pode ser visto pelas FROTAs deste classificador em todos os espaços, durante a seleção de características, apresentadas na Figura 4.5.

O melhor desempenho dos dois classificadores TM foram obtidos com o menor espaço de características, demonstrando a fraqueza do algoritmo em problemas de grandes dimensões, que pode ser observada claramente nas Figuras 4.10 e 4.11.

A melhor precisão obtida pelo k - NN é apenas 0,5% menor que a taxa média de acerto obtida pela MLP. Pode-se ainda dizer que a robustez dos dois classificadores, k - NN com distância *city block* e $k = 1$ e MLP, é semelhante. A grande diferença é o ICC, indicando que o k - NN é aproximadamente 9 vezes mais custoso computacionalmente que a MLP.

Enquanto a precisão dos classificadores TM diminui com o aumento da quantidade de características, que ocorre o oposto nas MLP, conforme visto na Figura 4.12. A diminuição da precisão nas MLP só ocorre no espaço S_5 , que considera um momento principal, o que diminui a qualidade do classificador.

A média da taxa de acerto dos melhores resultados das máquinas de vetores de suporte é de 95,12%, indicando grande precisão desta técnica. A média é mais alta que a do k - NN e do TM . Porém, o desvio padrão indica que, apesar de ser mais precisa que os métodos clássicos, as SVMs são, para estas configurações e condições de treinamento sem otimizações e testes, menos robustas que k - NN e TM .

A PT-SVM obteve um resultado inferior às outras técnicas de SVM. Todavia, ela apresenta a menor complexidade dentre os métodos que atingiram uma taxa média de acerto de no mínimo 90%. A escolha da melhor PT-SVM foi baseada na maior robustez, indicada pelo menor desvio padrão, pois as taxas médias de acerto para os espaços S_2 e S_3 são iguais, com desvios padrão de 1,63 e 1,72, respectivamente. A PT-SVM escolhida tem ICC igual a 331. Conforme apresentado na tabela 4.16, a complexidade mínima desta árvore otimizada por probabilidade é de 295, enquanto a máxima é de 729. Mesmo considerando o pior caso, esta técnica ainda obteve custo computacional melhor que as outras duas técnicas baseadas em SVM.

5 CONCLUSÕES

Este trabalho apresentou uma metodologia para resolução do problema de categorização de veículos terrestres baseada em visão computacional. As definições das etapas de extração e seleção de características e classificação foram propostas e avaliadas com base na estimativa da qualidade final do sistema e de sua complexidade computacional.

A qualidade foi avaliada através de um método baseado em testes exaustivos, que, com base em u conjuntos de treino e teste, implementa e avalia a performance de u classificadores. Além de calcular a taxa média de acerto e o desvio padrão, esta série de testes também foi utilizada para calcular a frequência relativa de ocorrência da taxa de acerto (FROTA), que é um histograma normalizado das taxas de acerto, cuja envoltória pode ser aproximada por uma Gaussiana, o que permite uma análise visual de precisão e robustez do classificador.

A complexidade computacional dos algoritmos envolvidos foi estimada por meio de um índice de complexidade computacional (ICC) que atribui pesos para diversas operações lógicas e aritméticas. O uso desta métrica nos possibilitou executar a estimação, de forma precisa, do custo computacional de cada um dos classificadores, e nos permitiu fazer a escolha do classificador ideal também por sua complexidade, e não somente por sua precisão e robustez.

As características utilizadas foram os momentos principais e os momentos invariantes à escala e translação. Após a etapa de seleção de características, que ordenou as 18 características extraídas da imagem através do algoritmo *SFFS*, pôde-se perceber claramente a superioridade dos momentos invariantes. As nove melhores características são momentos invariantes. O único momento principal que participou da avaliação de classificadores foi $\overline{m_{12}}$, no espaço de características S_5 , composto por este momento principal e todos os momentos invariantes.

Mesmo sendo incluído na avaliação de dez classificadores, o S_5 não figurou na lista dos melhores conjuntos espaço/classificador (tabela 4.21), mostrando que momentos centrais não são significantes à solução do problema proposto

Pode-se concluir que, com as taxas médias de acerto atingindo entre 69,42% para classificador *TM* com distância *city block* e espaço de características S_3 e 97,20% atingido pela MLP com espaço de características S_4 (que contém os nove momentos invariantes). As características propostas neste trabalho, além de fácil extração e baixo custo, possuem boa significância para a solução do problema de categorização de veículos nas quatro classes propostas.

5.1 TRABALHOS FUTUROS

Pela própria natureza do problema abordado, um caminho claro a ser seguido é o aumento e a divisão de classes. Para diferentes tipos de estudo de tráfego, diferentes categorias são requeridas. Para estudos de viabilidade de concessão de pedágios, uma métrica importante é a quantidade de eixos presentes em caminhões. Para análise de grandes metrópoles, como São Paulo, a inclusão da classe moto pode ser

necessária. Logo, o estudo e a implementação de técnicas que permitam a categorização em mais classes e sub-classes se fazem necessário.

Com relação à seleção de características, o uso de outros classificadores é recomendado para obtenção de resultados mais coerentes. Para a medição de desempenho, sugere-se o uso de conjuntos de tamanhos diferentes em uma fase inicial do problema, a fim de que se obtenha o menor conjunto u que seja capaz de caracterizar o problema.

Para aumentar a gama de classificadores considerados, pode-se utilizar outros algoritmos de treinamento para as redes neurais artificiais assim como utilizar diferentes *kernels* nas *SVMs*. Porém, o aumento do conjunto de classificadores deve ser levado em conta no cálculo do tempo necessário para efetuar o seu treinamento.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] PINTO, A. B. et al. Matriz de deslocamentos a partir de amostra de leitura de placas e contagem volumétrica em campo. In: *X ANPET*. [S.l.: s.n.], 1996.
- [2] JI, P.; JIN, L.; LI, X. Vision-based vehicle type classification using partial gabor filter bank. In: *2007 IEEE International Conference on Automation and Logistics*. [S.l.: s.n.], 2007. p. 1037–1040.
- [3] HUANG, C.; LIAO, W. A vision-based vehicle identification system. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. [S.l.: s.n.], 2004. v. 4.
- [4] GAJDA, J. et al. A vehicle classification based on inductive loop detectors. In: *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*. [S.l.: s.n.], 2001. v. 1, p. 460–464 vol.1.
- [5] SUN, C.; RITCHIE, S.; OH, S. Inductive classifying artificial network for vehicle type categorization. In: *Computer2013Aided Civil and Infrastructure Engineering*. [S.l.: s.n.], 2003. v. 18, n. 3, p. 161–172.
- [6] SHIN, P. et al. Automatic vehicle type classification using strain gauge sensors. In: *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*. [S.l.: s.n.], 2007. p. 425–428.
- [7] GUPTE, S. et al. Detection and classification of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, v. 3, n. 1, p. 37–47, 2002.
- [8] XIONG, N. et al. A Neural Network Based Vehicle Classification System for Pervasive Smart Road Security. *Journal of Universal Computer Science*, v. 15, n. 5, p. 1119–1142, 2009.
- [9] MORRIS, B.; TRIVEDI, M. Improved vehicle classification in long traffic video by cooperating tracker and classifier modules. In: *IEEE International Conference on Video and Signal Based Surveillance, 2006. AVSS'06*. [S.l.: s.n.], 2006. p. 9–9.
- [10] MUKUNDAN, R.; RAMAKRISHNAN, K. *Moment functions in image analysis: theory and applications*. [S.l.]: World Scientific, 1998.
- [11] HU, M. Visual pattern recognition by moment invariants. *Information Theory, IEEE Transactions on*, v. 8, n. 2, p. 179–187, 1962.
- [12] KAN, C.; SRINATH, M. Invariant character recognition with zernike and orthogonal fourier–mellin moments. *Pattern Recognition*, Elsevier, v. 35, n. 1, p. 143–154, 2002.
- [13] HADDADNIA, J.; FAEZ, K.; MOALLEM, P. Neural network based face recognition with moment invariants. In: *Image Processing, 2001. Proceedings. 2001 International Conference on*. [S.l.: s.n.], 2001. v. 1.
- [14] TEH, C.-H.; CHIN, R. On image analysis by the methods of moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 10, n. 4, p. 496–513, Jul 1988.

- [15] TORRES-MENDEZ, L. et al. Translation, rotation, and scale-invariant object recognition. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, v. 30, n. 1, p. 125–130, Feb 2000.
- [16] RAUDYS, S.; JAIN, A. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence*, v. 13, n. 3, p. 252–264, 1991.
- [17] BRAGATTO, T.; RUAS, G.; LAMAR, M. Real-time hand postures recognition using low computational complexity artificial neural networks and support vector machines. In: *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on*. [S.l.: s.n.], 2008. p. 1530–1535.
- [18] JAIN, A.; ZONGKER, D. Feature selection: evaluation, application, and small sample performance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 19, n. 2, p. 153–158, Feb 1997. ISSN 0162-8828.
- [19] PUDIL, P. et al. Floating search methods for feature selection with nonmonotonic criterion functions. In: *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on*. [S.l.: s.n.], 1994. v. 2, p. 279–283.
- [20] FUKUNAGE, K.; NARENDRA, P. A branch and bound algorithm for computing k-nearest neighbors. *Computers, IEEE Transactions on*, v. 100, n. 24, p. 750–753, 1975.
- [21] HAYKIN, S. *Neural Networks and Learning Machines*. [S.l.: s.n.], 1988.
- [22] VAPNIK, V.; LERNER, A. Pattern recognition using generalized portrait method. *Automation and Remote Control*, v. 24, p. 774–780, 1963.
- [23] BOSER, B. E. I. M. G.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. [S.l.: s.n.], 1992.
- [24] AIZERMAN, M.; BRAVERMAN, E.; ROZONOER, L. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, v. 25, p. 821–837, 1964.
- [25] HSU, C.; LIN, C. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, v. 13, n. 2, p. 415–425, 2002.
- [26] RIFKIN, R.; KLAUTAU, A. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, MIT Press Cambridge, MA, USA, v. 5, p. 101–141, 2004.
- [27] FÜRNKRANZ, J. Round robin classification. *The Journal of Machine Learning Research*, MIT Press Cambridge, MA, USA, v. 2, p. 721–747, 2002.
- [28] DIETTERICH, T.; BAKIRI, G. Solving multiclass learning problems via error-correcting output codes. *Arxiv preprint cs.AI/9501101*, 1995.
- [29] CHEONG, S.; OH, S.; LEE, S. Support vector machines with binary tree architecture for multi-class classification. *Neural Information Processing-Letters and Reviews*, v. 2, n. 3, p. 47–51, 2004.

- [30] CESA-BIANCHI, N.; GENTILE, C.; ZANIBONI, L. Hierarchical classification: combining Bayes with SVM. In: ACM NEW YORK, NY, USA. *Proceedings of the 23rd international conference on Machine learning*. [S.l.], 2006. p. 177–184.
- [31] Departamento Nacional de Trânsito. *Relatório da Frota Nacional de Veículos Terrestres – Maio*. 2009.