

Jadson Castro Gertrudes

***Utilização de classificador na detecção de veículos
em imagens para auxílio na condução de veículos***

Vitória da Conquista- BA, Brasil

14 de janeiro de 2011

Jadson Castro Gertrudes

***Utilização de classificador na detecção de veículos
em imagens para auxílio na condução de veículos***

Monografia apresentada para obtenção do Grau
de Bacharel em Ciência da Computação pela
Universidade Estadual do Sudoeste da Bahia.

Orientador:

Roque Mendes Prado Trindade

DEPARTAMENTO DE CIÊNCIAS EXATAS - DCE
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DA BAHIA - UESB

Vitória da Conquista- BA, Brasil

14 de janeiro de 2011

Monografia de Projeto Final de Graduação sob o título “*Utilização de classificador na detecção de veículos em imagens para auxílio na condução de veículos*”, defendida por Jadson Castro Gertrudes e aprovada em 14 de janeiro de 2011, em Vitória da Conquista, Estado da Bahia, pela banca examinadora constituída pelos professores:

Prof. Dr. Roque Mendes Prado Trindade
Orientador

Prof. Dr. Ivanor Nunes de Oliveira
Universidade Estadual do Sudoeste da Bahia

Profa. Dra. Alzira Ferreira da Silva
Núcleo de Tecnologia Educacional

Resumo

A criação de um veículo de fácil condução é tentada desde seu surgimento até hoje. O maior desejo de todos é que o condutor tenha menor preocupação no momento em que estiver a frente da direção. Diversos dispositivos são criados com a finalidade de manter esta tranquilidade aos condutores de veículos, dentre eles podemos citar airbag, cambio automatico, GPS, direção hidráulica, freios ABS, amortecedores, entre outros. Apesar do grande número de itens criados, ainda é grande o número de pessoas mortas no trânsito devido a diversas causas que diminuem a atenção do homem no trânsito. A visão computacional surge como um mecanismo de ajuda para criação de sistemas inteligentes para veículos, uma vez que uma das suas tarefas é “enxergar” situações e procurar as melhores soluções para elas. O presente trabalho tem uma pequena contribuição do uso da visão computacional para o auxílio na condução de veículos. O principal foco deste trabalho foi realizar aprendizado de máquina para detecção da parte traseira de carros.

Abstract

The creation of a vehicle within easy driving is attempted from its inception until today. The largest desire of all is that the driver has less concern at the moment that is the opposite direction. Several devices are created in order to maintain this tranquility of vehicle drivers, among them we can mention airbag, automatic transmission, GPS, power steering, ABS brakes, shock absorbers, among others. Despite the integral part number of items created, there is a huge number of people killed in traffic due to various causes that diminish the man's attention in traffic. Computer vision emerges as a mechanism help establish systems for intelligent vehicles, since one of its tasks is to "see" situations and seek better solutions for them. This work has a small exhibition of the use of computer vision to aid the driver in drive vehicles. The main focus of this study was to machine learning detection of the back of cars.

Dedicatória

Dedico este trabalho à minha família e aos meus amigos sem os quais não estaria apresentando este trabalho.

Agradecimentos

Agradeço primeiramente a Deus, apesar de não ter ido muito a igreja, pois sei que nos momentos mais difíceis da nossa vida recorremos a ELE.

Agradeço a minha mãe Josenilza, meu pai Gonçalo, e meus dois irmãos, Jardel e Josiney, pelo apoio em todas as fases da minha vida.

Agradeço a minha namorada Jessica pela compreensão durante este a realização da monografia.

E é claro que não iria deixar de agradecer a todos os meus amigos que me ajudaram nesta caminhada. (Tentei colocar em ordem cronológica, não sei se vai ficar bom não!!): Rafael, Kleyton, Diego, Bertoni, Bertonne, Pedro, Sílvia, Neylor, Hesdras, Luiz Hilário, Diogo, Douglésia, Elias, Walter, Gabriel, Ramon, Dino Sinthia, TODOS os moradores da Casa do Estudante de Livramento e muitos outros que sempre me deram apoio para estudar.

Não deixaria de homenagear o anjo do curso de Ciência da Computação, nossa amiga Celina, sempre prestativa mesmo nos momentos de estresse dela.

Agradeço também ao professor Roque Trindade, que tem ajudado muitos alunos desde sua conclusão do doutorado, mesmo com milhões de compromissos.

Lista de Figuras

1.1	Acidentes de trânsito, ano 2008	p. 13
2.1	Exemplos dos primeiros veículos	p. 16
2.2	Primeiros veículos produzidos em larga escala	p. 17
2.3	Automóveis na Primeira Guerra Mundial	p. 17
2.4	Airbag	p. 18
2.5	Freios ABS	p. 19
2.6	Amortecedor	p. 20
2.7	Posicionamento dos satélites no globo para funcionamento do GPS	p. 20
2.8	Funcionamento da direção hidráulica	p. 21
2.9	Modelo de câmbio automático	p. 21
2.10	Campo de atuação da visão computacional	p. 23
2.11	Exemplo de fractal	p. 25
2.12	Método de execução do algoritmo AdaBoost	p. 29
3.1	Captura das imagens para treinamento do classificador	p. 31
3.2	Rotular imagens utilizando o programa Positives Builder	p. 32
3.3	Dados rotulados após a escolha da região de interesse	p. 32
3.4	Extração de características	p. 33
3.5	Início do treinamento do classificador	p. 33
3.6	Término do treinamento do classificador	p. 34
3.7	Primeira tentativa de detecção dos veículos	p. 34
3.8	Última tentativa de detecção dos veículos	p. 35

Lista de Tabelas

3.1	Taxa de detecção de veículos (%)	p. 35
-----	--	-------

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 11
1.1	Contextualização e motivação	p. 11
1.2	Objetivos	p. 12
1.3	Justificativa	p. 12
1.4	Metodologia	p. 13
1.4.1	Classificação da pesquisa	p. 13
1.4.2	Tecnologias utilizadas	p. 14
1.5	Estrutura do documento	p. 15
2	Veículos e visão computacional	p. 16
2.1	Veículos	p. 16
2.1.1	Do surgimento do veículo até hoje	p. 16
2.1.2	Dispositivos desenvolvidos e adaptados para melhoria da segurança na direção	p. 18
2.2	Visão computacional	p. 22
2.2.1	Disciplinas aplicáveis	p. 22
2.2.2	Áreas de pesquisa	p. 24
2.2.3	Campos de aplicação	p. 26
2.2.4	Aprendizado de máquina	p. 27

3	Estudo de caso - Detecção de veículos em imagens	p. 30
3.1	Definição do objeto e coleta de dados	p. 30
3.2	Rotular dados	p. 31
3.3	Extração de características, treinamento e validação	p. 31
3.4	Testes	p. 33
4	Conclusões e trabalhos futuros	p. 36
4.1	Conclusões	p. 36
4.2	Trabalhos Futuros	p. 37
	Referências Bibliográficas	p. 38
	Apêndice A – Código fonte gerado pela ferramenta Harpia	p. 42

1 Introdução

1.1 Contextualização e motivação

Não há um momento exato para o surgimento do primeiro automóvel. Porém, desde o seu surgimento, muitas transformações ocorreram. Muitos destes automóveis que foram frutos de sucessivas aproximações e adaptações tecnológicas que, com o passar do tempo estão se desenvolvendo para um objetivo comum: viajar rápido, com comodidade, com o mínimo de esforço para os ocupantes e com o máximo de segurança.

Para auxiliar neste processo de desenvolvimento temos a visão computacional. Ela é a ciência e a tecnologia das máquinas que enxergam, onde "enxergar" significa que a máquina está habilitada a retirar informação necessária de uma imagem para resolução de problemas. Como uma disciplina científica, visão computacional está concentrada na teoria de sistemas artificiais que realizam extração de informações de imagens.

A teoria gerada pela visão computacional é aplicada na construção e modelagem de sistemas. Exemplos de aplicações como controle de processos (robôs industriais e veículos autônomos), organização de informação e modelagem de objetos.

Uma tarefa típica da visão computacional é a detecção de objetos em imagens. Nesta tarefa um sistema verifica se determinado objeto (carro, face, pedestre, texto entre outros) estão contidos numa imagem.

Motivado pelo trabalho apresentado por Viola e Jones (2001) e pela frase: "Alguns homens vêem as coisas como são e dizem: Por quê? Outros sonham com as coisas como nunca foram e se perguntam: Por que não?", o seguinte projeto tem por objetivo utilizar de métodos de classificação para detecção de veículos em imagens para auxílio na condução de veículos.

Foi utilizado o algoritmo de aprendizado AdaBoost para criação do modelo para detecção da parte traseira de carros. O algoritmo que será apresentado nas próximas seções faz parte de uma biblioteca da linguagem de programação C++ (OpenCV) que será apresentada na seção 1.4.2.

1.2 Objetivos

O objetivo deste projeto é utilizar um método para detecção de objetos em imagens, em especial a detecção de veículos.

Uma característica importante na concepção de sistemas de visão computacional é o nível de significância de teste e experimentação que normalmente é necessário antes de chegar a uma solução aceitável. Esta característica implica na capacidade de formular abordagens e criar rapidamente protótipos candidatos a soluções em geral.

O conteúdo apresentado neste trabalho tem como propósito essencial destiná-los aos docentes e discentes no curso de Ciências da Computação e para outras pessoas que possuam interesse no campo de visão computacional ou que queiram adquirir um pouco de informação sobre o tema. Também é de grande valia para quem queira conhecer um pouco mais sobre automóveis e de algumas tecnologias que fazem parte da sua composição.

1.3 Justificativa

Apesar da quantidade de adaptadores existentes que auxiliam a direção do motorista ainda é grande o número de acidentes de trânsito, muitos destes ocorridos devido a falhas do condutor. Falhas estas que podem ser velocidade excessiva, direção sob efeito de bebida alcoólica, desrespeito à sinalização, entre outros.

Dados e fontes encontrados em UNESP (2010) relatam que para que um veículo seja guiado com segurança é necessário que o ser humano esteja em boas condições técnicas e emocionais. Estes fatores influenciam no tempo de reação à uma situação perigosa, levando a causa de acidentes. No ano de 2008, segundo o DENATRAN, houve 428.970 acidentes no país (figura 1.1) muitos destes com vítimas fatais.

Reconhecimento de objetos em visão computacional é a tarefa de encontrar um dado objeto numa imagem ou vídeo. Os seres humanos reconhecem um grande número de objetos em imagens com pouco esforço, apesar do fato de que a imagem dos objetos podem variar em pontos de vista, tamanho, escala ou mesmo quando são traduzidos ou rotacionados.

Objetos podem ser reconhecidos, mesmo quando estão parcialmente obstruídos do campo de visão. Esta tarefa ainda é um desafio para os sistemas de visão computacional. Zhou, Ma e Chelberg (2006) relatam que o sucesso para a detecção de objetos em imagens ocorrerá quando sistemas de visão computacional reconhecerem itens em ambientes sem estrutura e com iluminação dinâmica.

QUADRO 5 - ACIDENTES DE TRÂNSITO COM VÍTIMAS - POR UF													
UF	Total	Tipo de Acidente						Fase do Dia			Área		
		Colisão/Abalroamento	Tombamento/ Capotamento	Atropelamento	Choque c/ Objeto Fixo	Outros	Não Inform	Dia	Noite	Não Inform	Urbana	Rural	Não Inform
BRASIL	428.970	239.728	49.793	59.405	36.938	35.791	7.223	252.699	169.619	5.824	288.702	59.918	76.952
Acre	2.207	1.637	123	241	85	119	2	1.284	902	21	2.188	19	0
Alagoas()	1.868	1.159	179	183	149	188	2	1.107	753	0	1.128	732	0
Amapá	1.328	1.061	30	157	55	20	5	505	823	0	-	-	1.328
Amazonas	4.765	1.931	135	1.127	247	715	610	2.741	1.993	1	3.207	1.558	0
Bahia()	9.388	3.758	529	2.724	1.022	1.325	30	5.855	3.524	9	8.566	822	0
Ceará	9.499	6.149	442	1.714	409	633	343	5.879	3.756	55	7.452	907	1.331
Distrito Federal()	18.393	7.147	308	1.867	458	613	-	6.462	3.931	-	10.393	-	-
Espírito Santo	14.869	7.967	1.748	1.733	1.480	1.125	16	8.583	5.428	58	10.166	3.509	394
Goiás	79.824	47.294	17.405	7.357	5.238	2.530	0	46.250	32.748	28	68.152	8.156	178
Maranhão***	2.883	1.272	192	337	81	201	0	1.199	884	0	1.290	780	13
Mato Grosso***	6.436	543	168	58	27	327	5.313	648	473	5.315	22	1.098	5.316
Mato Grosso do Sul	7.611	5.617	219	830	667	544	34	5.804	2.107	0	7.084	546	11
Minas Gerais****()	7.619	4.404	150	1.533	767	765	0	5.283	2.330	6	7.619	0	0
Pará	11.489	6.340	615	2.353	198	1.983	0	7.390	4.099	0	7.337	778	3.374
Paraíba**	2.915	1.555	233	415	265	291	156	1.711	1.204	0	2.308	607	0
Paraná	45.631	26.469	2.965	4.895	3.671	7.625	6	26.395	19.228	8	34.042	7.948	3.641
Pernambuco()	2.777	2.018	51	384	168	54	10	1.857	919	1	2.747	29	1
Piauí	3.522	2.455	171	247	145	504	0	2.063	1.459	0	2.910	612	0
Rio de Janeiro()	26.300	6.238	684	5.003	975	7.400	-	12.081	8.215	4	1.630	1.128	17.542
Rio Grande do Norte	4.198	2.788	338	488	249	331	4	2.520	1.676	2	3.295	598	305
Rio Grande do Sul()	42.104	21.855	6.426	3.660	8.625	1.538	-	26.638	15.466	-	-	-	42.104
Rondonia	9.941	6.323	1.403	865	376	503	467	5.995	3.692	234	7.278	1.267	1.396
Roraima***	1.846	1.476	41	106	27	196	0	1.144	702	0	1.714	132	0
Santa Catarina	4.797	3.042	642	464	303	343	3	2.992	1.800	5	3.521	1.264	12
São Paulo()	115.414	64.956	13.983	20.295	10.822	5.159	199	66.540	48.823	51	89.435	25.979	0
Sergipe()	1.383	1.139	49	88	49	55	3	948	435	0	1.106	277	0
Tocantins	5.380	3.133	562	581	380	704	20	3.125	2.249	6	4.202	1.172	6

Fonte: Detrans.

(Vide Guia Observações Estatísticas

*** MG: Jan a JUN de 2008

*** PE: Semestre Recolto - Ita de Itamaracá e Petrolina (Jan a Jul)

Figura 1.1: Acidentes de trânsito, ano 2008

Fonte: DENATRAN

É cada vez maior o número de sistemas que utilizam visão computacional. Sistemas para detecção de placa de veículos (LENSKY; JO; GUBAREV, 2006), detecção de veículos (JIN; DAVIS, 2007), reconhecimento de faces (VIOLA; JONES, 2001), detecção de doenças em imagens médicas (NUNES, 2006; JIN et al., 2009) e sistemas de vigilância são alguns exemplos. Estudo para sistemas para assistência a deficientes (FRERE; DANGEL; CORDEIRO, 2001) também estão sendo realizados neste campo. Devido ao elevado número de aplicações, o estudo de algoritmos, metodologias são cada vez mais requisitados.

1.4 Metodologia

1.4.1 Classificação da pesquisa

O presente projeto tem em seu objetivo inicial capturar o máximo de informações sobre o tema proposto, garantindo assim a elaboração de uma pesquisa bibliográfica contendo as principais características. Essa pesquisa define uma maior ligação entre o que foi examinado ao que foi escolhido como tema do projeto. A pesquisa bibliográfica permite ao investigador uma gama de fenômenos do que aquela que poderia ser adquirida pesquisando diretamente.

Outro objetivo deste projeto foi a elaboração de um estudo de caso. Segundo Gil (2002)

o estudo de caso é encarado como o delineamento mais adequado para a investigação de um fenômeno contemporâneo dentro de seu contexto real.

1.4.2 Tecnologias utilizadas

Na criação dos arquivos para detecção dos veículos foi utilizada a biblioteca OpenCV. Ela é uma biblioteca de funções para visão computacional em tempo real escrita em C e possui vantagens para processamento em máquinas com múltiplos processadores. OpenCV é uma biblioteca famosa e suas principais comunidades estão na China, Japão, Rússia, Europa e Israel.

Uma meta desta biblioteca é fornecer uma infraestrutura de aplicação de uso simples para auxiliar no desenvolvimento rápido de aplicações de visão computacional. Mais detalhes sobre esta biblioteca podem ser encontradas em (BRADSKY; KAEHER, 2008).

Na visualização dos resultados foi utilizada a ferramenta Harpia (S2I, 2010). Esta ferramenta foi desenvolvida pelo grupo de Sistemas Industriais inteligentes da Universidade Federal de Santa Catarina. Tem em seu objetivo principal a construção de um ambiente gráfico para aprendizado, implementação e gerenciamento de sistemas de visão computacional.

O sistema foi feito para vários módulos de software, para comunicação de hardware, processamento de sinais e gerenciamento remoto de sistemas de visão.

Para formatação deste trabalho foi utilizado LaTeX. Ela é um conjunto de macros para produção de textos matemáticos e científicos. Também é utilizado para produção de cartas pessoais, artigos e livros (WIKIPEDIA, 2010c).

A idéia principal do LaTeX é distanciar o autor o máximo possível da apresentação visual da informação, pois a constante preocupação com a formatação desvia o pensamento do autor do conteúdo do documento. O usuário é encorajado a trabalhar com conceitos mais lógicos.

As principais vantagens no uso da LaTeX são:

- O algoritmo avançado do TeX, que permite a criação de documentos com aparência profissional.
- A edição de fórmulas matemáticas.
- Estruturas tipográficas como bibliografia, tabela de conteúdo são geradas facilmente.
- São programas livres.

1.5 Estrutura do documento

O projeto está organizado na seguinte forma: O capítulo 2 apresenta algumas características de veículos automotores e conceitos da visão computacional.

No capítulo 3 temos a apresentação do trabalho realizado (detecção da parte traseira de veículos utilizando classificador).

No capítulo 4 temos a conclusão do trabalho além da descrição de trabalhos futuros.

Por fim, a última sessão irá conter as referências bibliográficas utilizadas na confecção deste trabalho.

2 *Veículos e visão computacional*

2.1 Veículos

2.1.1 Do surgimento do veículo até hoje

Milhares de anos separaram a invenção da roda dos primeiros veículos autopropulsores. Durante centenas de anos o mundo testemunhou a criação de veículos movidos a vento, força humana ou força animal (figura 2.1). De fato, essa história iniciou após a criação dos motores movidos a vapor e que eram capazes de realizar o transporte humano (ECKERMANN, 2001).

A história do automóvel começa aproximadamente em 1769. Em 1806, os primeiros carros



Figura 2.1: Exemplos dos primeiros veículos
imagem da esquerda - Veículo movido a vento; imagem central - Veículos movidos a força humana; imagem à direita - veículo movido a força animal. Fonte: (ECKERMANN, 2001)

movidos por motores de combustão interna a gás apareceram, o que levou à introdução no ano de 1885 do motor de combustão modernos a gasolina. Carros movidos a electricidade apareceram na virada do século 20, mas praticamente desapareceram do comum até a virada do século 21, segundo Hollanda (2010).

Como tal, o início da história do automóvel pode ser dividido em um número de épocas com base no método predominante de propulsão automotiva durante esse tempo. Mais tarde, períodos foram definidos por tendências de estilo exterior e tamanho e preferências de utili-

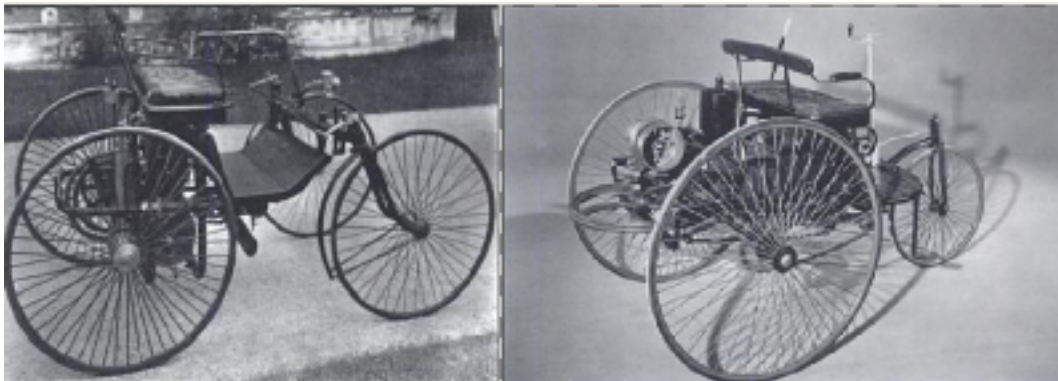


Figura 2.2: Primeiros veículos produzidos em larga escala
imagem da esquerda - veículo criado por Daimler; imagem à direita - veículo criado por Karl Benz para produção em massa. Fonte: (ECKERMANN, 2001)



Figura 2.3: Automóveis na Primeira Guerra Mundial
Fonte: (ECKERMANN, 2001, p 73)

dade.

A primeira época do automóvel foi definida entre os anos de 1880 e 1920. Esta fase foi marcada pelo desenvolvimento dos primeiros veículos com motores de combustão, criação de veículos por Gottlieb Daimler e Wilhelm Maybach e pela primeira linha de produção de por Karl Benz (figura 2.2). É marcada também pelo início da produção de veículos por todo o mundo, surgimento de diversas empresas especializadas em construção de veículos (FIAT, PEUGEOT, FORD, entre outras) e pelo uso dos automóveis durante a Primeira Guerra Mundial (1914-1918), figura 2.3.

O segundo período dos automóveis (1920 - 1980) é apresentado pela grande venda de veículos fechados e pelo rápido desenvolvimento dos motores a combustão, de acordo a Georganio (2003). Outros fatos foram a Segunda Guerra Mundial (1939 - 1945), a criação de um dos mais famosos campeonatos de corrida, a Fórmula 1, após a guerra e também um aumento na competição para venda de veículos.

No campo da tecnologia teve um crescente foco na concepção de segurança de automóveis,

e na utilização de combustíveis alternativos, principalmente após a crise do petróleo em 1973.

A era moderna, ou terceiro período, é definida desde a década de 80 até a corrente data. Esta época diferencia da antiga principalmente pela melhoria no design dos veículos e pelo auxílio do computador na criação de projetos. Também houve um rápido aumento da eficiência dos combustíveis e da potência do motor através do controle por sistemas informatizados e da preocupação com a segurança do motorista. Para isto diversos dispositivos foram e estão em desenvolvimento.

2.1.2 Dispositivos desenvolvidos e adaptados para melhoria da segurança na direção

Os dispositivos de segurança acoplados aos veículos auxiliam tanto na condução quanto na proteção dos ocupantes. São apresentados alguns destes dispositivos que ajudam na redução de vítimas fatais no trânsito.

O **Airbag** é um dispositivo de segurança do automóvel (figura 2.4). É projetado para evitar que os ocupantes do veículo golpeie algum objeto (volante, painel, etc) no caso de ocorrência de colisão do veículo. Este aparelho foi desenvolvido por Jhon Hetrick em 1952 durante realização de experimentos na marinha dos Estados Unidos.

O seu funcionamento dá-se através do monitoramento de sensores e giroscópios, que en-



Figura 2.4: Airbag
Fonte: (AUTOPÉDIA, 2010)

viam informações para a unidade de controle do airbag. Atendido ou excedido o limite destes sensores, o controle do airbag irá acionar um propulsor para inflar o saco de nylon com gás nitrogênio (N_2).

Os **freios** também são outras fontes de segurança. Os freios ABS (Anti-block System), figura 2.5, são os mais utilizados, pois permitem ao condutor o controle total do veículo no

momento da frenagem e também pode reduzir a distância de frenagem em algumas superfícies escorregadias (BURTON et al., 2004).

O freio ABS típico inclui uma unidade de controle central eletrônica (ECU), quatro sen-



Figura 2.5: Freios ABS
Fonte: do frotista (2010)

sores de velocidade das rodas, e pelo menos válvulas hidráulicas no sistema hidráulico do freio. O ECU monitora constantemente a velocidade de rotação de cada roda e em caso de frenagem ela aplica sobre os freios o efeito de “liberar e pressionar“, evitando o bloqueio da roda. Caso ocorra uma falha em qualquer parte do ABS, uma luz de advertência geralmente será acesa no painel de instrumentos do veículo, e o ABS será desativado até que a falha seja corrigida. O ABS moderno aplica uma pressão de freio individuais para as quatro rodas através de um sistema de controle de hub-sensores montados e um dedicado micro-controlador.

O ABS é oferecido ou é padrão na maioria dos veículos rodoviários produzidos hoje. O equipamento ABS também pode ser usado para implementar um sistema de controle de tração.

O **amortecedor** entra nesta lista de dispositivos como um item tanto de conforto quanto de segurança. Ele é um dispositivo mecânico criado para manter o contato das rodas com o chão e também dissipar ou absorver a energia gerada por elas (figura 2.6) .

Um tema recente deste dispositivo é a utilização da energia gerada por ele na utilização do próprio veículo (GEOGRAPHIC, 2010). Isto ajuda a diminuir o uso de combustível e sucessivamente a melhoria do meio ambiente.

O Sistema de Posicionamento Global (**GPS**) é baseado em um sistema de navegação global por satélite, que fornece informações fiéis de localização, velocidade e tempo, em qualquer tempo e lugar da Terra. Ele é mantido pelo governo dos Estados Unidos e é de livre acesso por qualquer pessoa que possua um receptor GPS (ADMINISTRATION, 1995).

Embora tenha sido desenvolvido para fins militares, o GPS tem aplicações importantes para fins civis. É utilizado para fins agrícolas, atividades de telefonia, topografia, geografia, entre

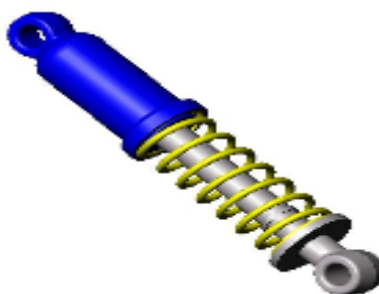


Figura 2.6: Amortecedor

Fonte: (UNIÃO, 2010)

outros.

Para os veículos deu a possibilidade de realizar o deslocamento independente de condutor. Muitos trabalhos já estão sendo realizados para este tipo de condução. Alguns trabalhos desenvolvidos podem ser encontrados em (ESPORTE, 2010c), (ESPORTE, 2010a) e (ESPORTE, 2010b).

O funcionamento do GPS dá se através do envio e recepção de sinais enviados por satélites

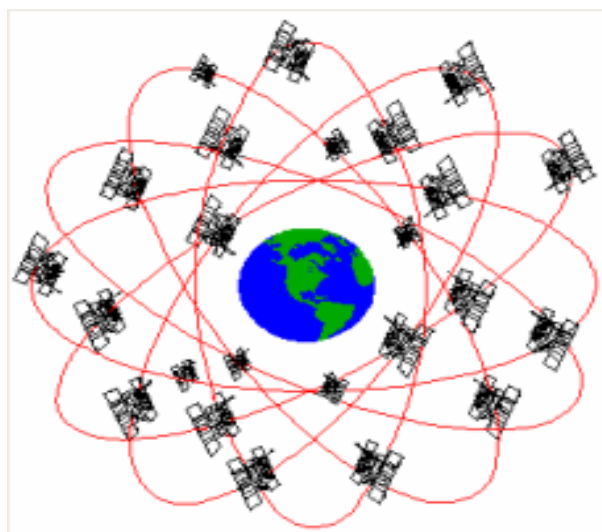


Figura 2.7: Posicionamento dos satélites no globo para funcionamento do GPS

Fonte: (CRUZEIROS, 2010)

situados na órbita terrestre aos dispositivos de posicionamento. Através de cálculos de distância de cada satélite e da posição do satélite na órbita é possível realizar o cálculo da posição com boa precisão.

Outro item fundamental para a segurança na direção é a utilização da **direção hidráulica**. De acordo Nunney (2006) sua primeira utilização em veículos foi feita na década de 20, para diminuir o peso dos veículos automotores que ganhavam cada vez maiores proporções.

O centro de um sistema convencional é um mecanismo de servo-hidráulica, que é posto em

funcionamento quando a resistência ao giro do volante excede uma quantidade predeterminada. Em seguida, envolve um esforço adicional para auxiliar na operação manual da direção e, portanto, reduz o esforço da direção, que seriam impostas ao condutor (figura 2.8).

Um item utilizado nos Estados Unidos desde 1950 é o **câmbio automático**, figura 2.9.



Figura 2.8: Funcionamento da direção hidráulica
Fonte: (HYDRAUSYSTEM, 2010)

Segundo Miller, Vandome e McBrewster (2009) uma transmissão automática é uma caixa de velocidades do automóvel que pode mudar relações de marchas automaticamente liberando o motorista de ter que mudar de marcha manualmente.

Ele funciona através da relação entre velocidade e a rotação do motor, propondo manter a rotação do motor constante. As vantagens de utilizar este equipamento são o conforto e segurança, além de manter uma maior durabilidade do motor do veículo.

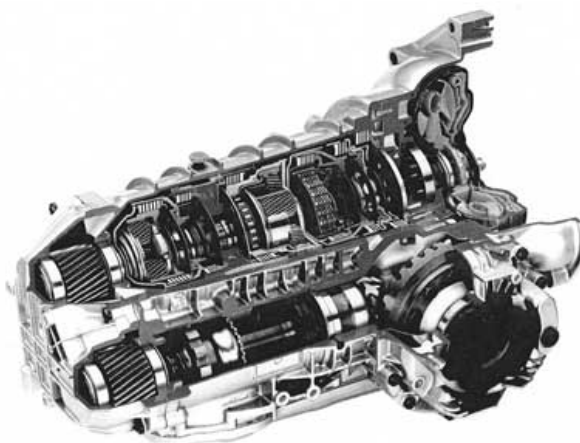


Figura 2.9: Modelo de câmbio automático
Fonte: (AUTOMÁTICO, 2010)

2.2 Visão computacional

Visão computacional é a transformação de dados adquiridos de uma imagem em qualquer decisão e todas estas transformações são realizadas para alcançar algum objetivo específico. Para toda esta transformação, diversas etapas podem ser necessárias ou não. Forsyth e Ponce (2002) dividem estes passos como: formação da imagem, modelos de imagens, visão antecipada, visão de nível médio e visão de alto nível. Abaixo temos uma abordagem geral deste campo, de acordo (FORSYTH; PONCE, 2002).

- **Formação da imagem:** aborda aspectos como formação da luz (luz no espaço, luz em superfícies, etc), fontes, sombras, sombreamento e cor.
- **Modelos de imagens:** descrição de características geométricas das imagens (curvas, superfícies, entre outras), características analíticas (geometria euclidiana, parâmetros de câmera, métodos de calibração) e probabilidade.
- **Visão antecipada:** trata de efeitos aplicados em imagens (filtros, detecção de bordas, textura e análise de múltiplas imagens).
- **Visão de nível médio:** como na visão antecipada, também trata de efeito sobre imagens (segmentação, montagem e movimento).
- **Visão de alto nível:** Nesta seção, são tratados aspectos de reconhecimento e classificação das imagens (encontrar modelos usando classificadores e reconhecimento por relações entre modelos). Este tópico será detalhado no decorrer do trabalho.

Segundo Trucco e Verri (1998) a visão computacional não se resume somente a estudo de algoritmos e criação software. Muitas ferramentas são necessárias para este tipo de sistema, incluindo hardware para aquisição e armazenamento das imagens, processamento das imagens e comunicação dos resultados para usuários ou sistemas automatizados. Outras áreas são fundamentais para o campo da visão computacional. A figura 2.10 nos dá uma dimensão da dependência da visão computacional com outros campos.

2.2.1 Disciplinas aplicáveis

A visão computacional envolve uma multidisciplinaridade de assuntos ao longo dos anos. Ela passa o campo da inteligência artificial, robótica, processamento de sinais, reconhecimento de padrões, psicologia, neurociência, e outros como ilustrado na figura 2.10. De acordo Trucco

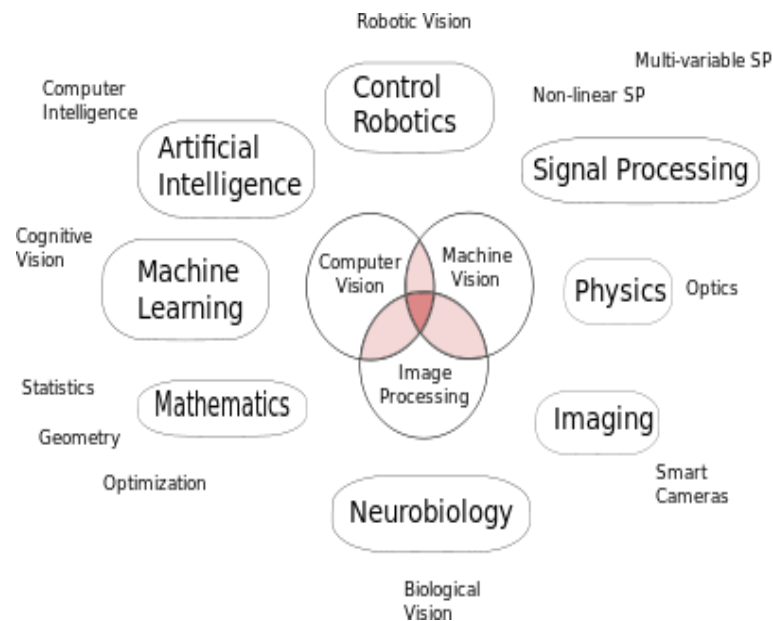


Figura 2.10: Campo de atuação da visão computacional
 Fonte: (WIKIPEDIA, 2010a)

e Verri (1998), como consequência deste vasto campo e do rápido crescimento a definição de escopo da visão computacional ainda estão em debate, de modo que todas as definições devem ser analisadas. Algumas disciplinas/áreas que estão estritamente relacionadas com a visão computacional são descritas a seguir:

- **Processamento de imagens:** É uma forma de processamento de sinal, onde a entrada é uma imagem (fotografia ou frame de vídeo) e a saída pode ser outra imagem ou uma série de características ou parâmetros descritos na imagem. Muitas técnicas de processamento de imagens envolvem tratamento de imagem em duas dimensões. Trucco e Verri (1998) relatam que esta área difere da visão computacional no que concerne propriedades das imagens e transformações de imagens para imagens, já que a visão computacional transforma imagens em dados para tomada de decisões. Exemplos de processamentos de imagens estão relacionados através do *enhancement* (computando uma imagem de melhor qualidade do que a original), compressão, restauração e extração de características.
- **Reconhecimento de padrões:** é a atribuição de algum tipo de valor de produção para um dado valor de entrada, de acordo com algum algoritmo específico. É classificado de acordo com o tipo de procedimento de aprendizagem utilizado para gerar o valor de saída. Por um longo tempo, reconhecimento de padrões produziu técnicas para reconhecimento

e classificação de objetos usando imagens digitais. Muitos métodos desenvolvidos no passado trabalham bem com objetos 2D ou 3D apresentados em diferentes poses.

- **Fotogrametria:** é definida como a arte de extrair de fotografias as métricas, a forma, as dimensões e a posição dos objetos nela contidos (TRUCCO; VERRI, 1998; WIKIPEDIA, 2010d). Ela utiliza métodos de disciplinas como óptica e geometria projetiva. A principal diferença desta área com as áreas de reconhecimento de padrões e processamento de imagens é que ela define altos níveis de precisão, caso que não ocorre na visão computacional.

2.2.2 Áreas de pesquisa

Um grande número de publicações no campo são realizados todos os anos. Muitos destes trabalhos podem ser encontrados em (XPLORE, 2010) e (ELSEVIER, 2010). Abaixo temos algumas áreas de pesquisa em visão computacional e trabalhos relacionados na área:

- **Deteção de características em imagens** - em visão computacional este conceito refere-se a métodos que visam na computação abstrações de informação de imagem e de tomada de decisões locais em cada ponto da imagem. Alguns trabalhos relacionados à área são:
 - Shao (1999) descreve em seu trabalho um modelo global de detecção de características em uma multi-imagem. A rede de multi-imagem é usada para descrever geometrias.
 - Ginige (1992) apresenta um algoritmo baseado em máquina de estados finitos (FSM) para detecção de linhas e bordas em imagens. Ela mostra que a aplicação deste método reduz o tempo de execução e melhora a qualidade das características analisadas.

Outros trabalhos realizados com detecção de características em imagens podem ser encontradas em (STOKMAN; GEVERS, 2007) e (MUYUN; MINGYI, 2006).

- **Estereo-fotometria** - é o processo de percepção visual que conduz a percepção de profundidade a partir de de duas projeções sobre a retina dos dois olhos. Na visão computacional é utilizada em robótica móvel para detectar obstáculos. Um exemplo de aplicação é (MOUNTNEY; STOYANOV; YANG, 2010) que faz uma descrição de visão estereo para aplicação em robótica cirúrgica.



Figura 2.11: Exemplo de fractal
Fonte: (WIKIPEDIA, 2010b)

- **Visão em cor** - é a capacidade de um organismo ou máquina para diferenciar objetos com base na frequência da luz que eles refletem, emitem ou transmitem. O sistema nervoso define cor através da comparação das respostas à luz dos vários tipos de cones fotorreceptores nos olhos (LUKAC; PLATANIOTIS, 2007). (UCHIKAWA, 1999) investiga em seu trabalho a relação entre memória e percepção da cor usando a memória de correspondência.
- **Reconhecimento por dimensão fractal** - O conceito de fractal foi introduzido em 1982 por B. Mandelbrot, para caracterizar figuras geométricas naturais que poderiam ser, por exemplo, ásperas ou irregulares. Segundo (KINSNER, 2005) um fractal é um fragmento geométrico que pode ser dividido em várias partes, onde cada uma destas partes representam uma cópia do objeto como um todo. A figura 2.11 mostra um exemplo de fractal. Barnsley (1988) descreve a geometria fractal como uma extensão da geometria clássica. Fala também que ela pode ser utilizada para realizar a modelagem precisa de objetos que variam desde samambaias até galáxias. Abaixo temos alguns trabalhos importantes que de fractal na detecção de objetos.
 - (LENSKY; JO; GUBAREV, 2006) utilizam o conceito de fractal para detecção de placas de veículos com boa taxa de detecção e eliminação de ruídos e perturbações.
 - (IDA, 1998) descreve em seu trabalho a utilização de fractal para segmentação e detecção de bordas dando ênfase para compressão de imagens.
 - (JIN et al., 2009) utilizam o método recurso multi-fractal aplicado em detecção de pequenas lesões em imagens geradas por raios-x.

Outros campos de pesquisa da visão computacional, de acordo Trucco e Verri (1998) são: Análise de movimento, detecção de objetos, detecção de objetos 3D, alta performance e arquiteturas em tempo real.

2.2.3 Campos de aplicação

Os estudos desenvolvidos na visão computacional são aplicados em diversas áreas. Abaixo temos a descrição e trabalhos relacionados a algumas áreas.

- **Inspeção industrial e controle de qualidade** - nesta área os sistemas são desenvolvidos para assegurar que produtos ou serviços sejam projetados e produzidos para ir ao encontro ou superar as expectativas dos usuários. Alguns trabalhos importantes para esta área são descritos por Kusano et al. (2008) e Konig, Genter e Glesner (1993).
- **Engenharia Reversa** - é o processo de descobrir os princípios tecnológicos de um dispositivo, objeto ou sistema através da análise de sua estrutura, função e operação. No campo da visão computacional podemos citar (THOMPSON et al., 1999) e (HO; CHUA, 1999) que aplicam técnicas de aquisição de imagens 3D para modelagem de objetos.
- **Reconhecimento de Faces** - aplicações computacionais que verificam a existência de uma pessoa numa foto ou video. (SCHNEIDERMAN; KANADE, 2000) e (VIOLA; JONES, 2001) demonstram técnicas para detecção de faces.
- **Controle de tráfego** - controle de tráfego é uma ocupação ao ar livre, dia e noite, por longas horas em todos os tempos e é considerada uma profissão perigosa, devido ao alto risco dos funcionários serem atingidos por veículos que passam. Nestas situações, sistemas de visão computacional são de grande importância. Alguns trabalhos realizados podem ser encontrados em (XIE et al., 2007; JIN; DAVIS, 2007; LENSKY; JO; GUBAREV, 2006; INIGO, 1989; LIU; PAYEUR, 2003).
- **Análise de Imagens Médicas** - Aplicações CAD (Computer Aided Diagnosis) são cada vez mais utilizadas nesta área. De acordo Giger (2000) CAD é um método naquele que o especialista médico utiliza os resultados de uma análise computadorizada de imagens médicas como uma "segunda opinião" na detecção de lesões e na elaboração do diagnóstico. De acordo Nunes (2006) aplicações com finalidades diversas tem sido desenvolvidas por vários grupos de pesquisas, visando a auxiliar na composição de diagnósticos como uma forma de contribuir para a detecção precoce de doenças. Trabalhos relacionados a

área são encontrados em (LO et al., 1995; JIN et al., 2009; YAP; EDIRISINGHE; BEZ, 2006; FRERE; DANGEL; CORDEIRO, 2001).

2.2.4 Aprendizado de máquina

A meta do aprendizado de máquina é transformar dados em informações (BRADSKY; KAEHER, 2008). De acordo Sebe et al. (2005) este tema adquiriu grande importância na visão computacional devido a criação de produtos comerciais a partir de técnicas de visão computacional e de uma tendência geral no campo de visão por computador para incorporar as técnicas de aprendizado em máquina. O aprendizado em máquina pode oferecer muitos benefícios ao campo da visão computacional, por (SEBE et al., 2005, p 2): Os algoritmos de aprendizado de máquina podem ser aplicados no campo de visão computacional para a melhoria da percepção do ambiente e na representação dos conhecimentos necessários ao sistema para executar sua tarefa. Em geral, os algoritmos de aprendizagem necessitam de vetores de entrada. Os tópicos abaixo apresentam as diversas etapas para execução destes algoritmos:

- **Definição do objeto** - nesta etapa é necessário a definição do objeto que será treinado. Como exemplo temos a aquisição de imagens de veículos: Um problema é como classificar estes dados (existe um carro ou não existe um carro). Os dados aparecem em diferentes formas, escalas e iluminação. É muito importante a definição “do que é um objeto“ numa determinada cena.
- **Coletar os dados** - situações de onde encontrar o objeto de estudo, analisar as suas variações (está em um local escuro ou claro). É preciso capturar as variações que ocorrem no objeto: diferentes tipos de luzes, condições do tempo, sombras, entre outros.
- **Classificação** - Após a aquisição dos dados é necessário rotulá-los (dizer se existe ou não um determinado objeto numa imagem). Supondo que uma pessoa tenha adquirido umas centenas de imagens, as estratégias são utilizar ferramentas que agilizam o processo ou procurar alguém que realize este tipo de serviço.
- **Extração de características** - Após rotular todas as imagens mais uma tarefa surge: decidir quais tipos de características serão retiradas do objeto. Se o objeto não possui variância não é preciso aplicar técnicas de processamento de imagens para adquirir outras características. Porém, alguns casos é preciso aplicar técnicas de processamento de imagens (rotação , redimensionamento, equalização de histograma, etc). O resultado desta extração é um vetor com cada rótulo associado a um objeto, cena ou ação.

- **Treinamento, validação e testes** - Uma vez que os dados são coletados é a vez de treiná-los. A utilização de frameworks podem auxiliar nesta fase.
- **Escolha do classificador a ser utilizado** - Por fim, depois que os dados são preparados você precisa escolher seu classificador. Para escolha deste classificador dados computacionais e de memória são determinísticos.

Classificador AdaBoost

Boosting é um algoritmo de aprendizado para realização de meta-aprendizado supervisionado. Ele é baseado na observação de que encontrando muitas regras "brutas" pode ser mais fácil do que encontrar uma regra simples, com alta predição (SCHAPIRE, 2003). Meir e Rätsch (2003) definem matematicamente como um conjunto de hipóteses h_1, h_2, \dots, h_T e considerando a composição do conjunto de hipóteses

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x). \quad (2.1)$$

Na equação α_t denota o coeficiente com que o conjunto h_t é combinado. O algoritmo boosting se refere a um método geral e comprovadamente eficaz de produzir regra de previsão, combinando regras ásperas e imprecisas.

AdaBoost (Adaptive Boosting) é o mais conhecido desta classe de algoritmos e foi descrito em 1995 para resolver problemas dos algoritmos de boosting (SCHAPIRE, 2003, p 3). Este algoritmo tem como entrada um conjunto de treino $(x_1, y_1), \dots, (x_m, y_m)$ onde cada x_i representa um domínio ou instância de espaço de X e cada rótulo y_i está em algum rótulo do conjunto Y . A figura 2.12 e o pseudocódigo deste algoritmo são apresentados (MEIR; RÄTSCH, 2003):

1. **Entrada:** $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$, número de iterações T
2. **Inicializa:** $d_n^{(1)} = \frac{1}{N}$ para todo $n = 1, \dots, N$
3. **Para cada** $t = 1, \dots, T$ faça,
 - (a) Treine o classificador com a respectiva amostra ponderada definida $\{S, d^{(t)}\}$ e obtenha hipóteses $h_t : x \mapsto \{-1, +1\}$,
 - (b) Calcule o erro ponderado no treinamento das hipóteses.

$$\epsilon_t = \sum_{n=1}^N d_n^{(t)} I(y_n \neq h_t(x_n)), \quad (2.2)$$

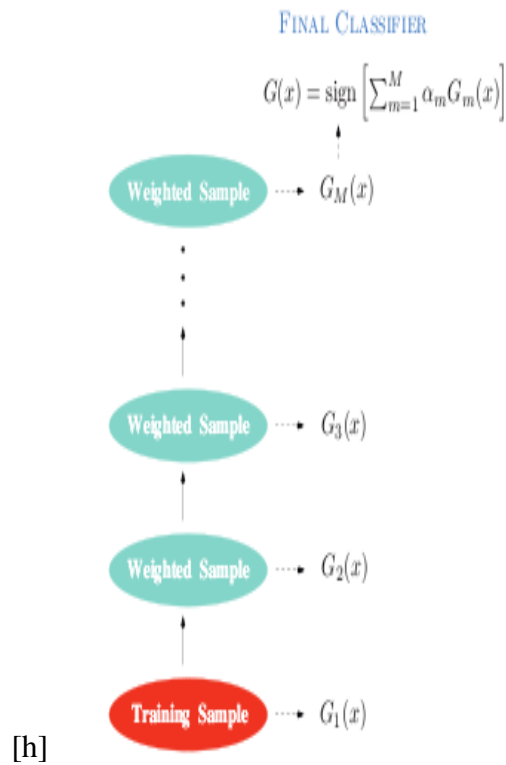


Figura 2.12: Método de execução do algoritmo AdaBoost
 Fonte: (HASTIE; TIBISHIRANI; FRIEDMAN, 2009, p 338)

(c) Ajuste:

$$\alpha_t = \frac{1}{2} \log \frac{1 - \varepsilon_t}{\varepsilon_t} \quad (2.3)$$

(d) Atualize os pesos:

$$d_n^{(t+1)} = \frac{d_n^{(t)} \exp(-\alpha_t y_n h_t(x_n))}{Z_t} \quad (2.4)$$

onde Z_t é uma constante de normalização tais que $\sum_{n=1}^N d_n^{(t+1)} = 1$.

4. **Pare se** $\varepsilon_t = 0$ ou $\varepsilon_t \geq \frac{1}{2}$ e ajuste $T = t - 1$.

5. **Saída:** $f_T(x) = \sum_{i=1}^T \frac{\alpha_i}{\sum_{r=1}^T \alpha_r} h_i(x)$

Recentemente técnicas de boosting estão tendo bastante sucesso em várias aplicações. A próxima seção contém a apresentação do estudo de caso utilizando esta técnica.

3 Estudo de caso - Detecção de veículos em imagens

Para o treinamento do classificador para detecção de veículos foi utilizado os processos para o aprendizado de máquina (descritos na seção 2.2.4).

3.1 Definição do objeto e coleta de dados

O objeto de estudo deste projeto foi a detecção da parte traseira de carros. A escolha da parte traseira foi feita pois o Código de Trânsito Brasileiro (BRASIL, 2002), artigo 29, Inciso III, dispõe que o condutor deverá guardar a distância de segurança lateral e frontal entre o seu e os demais veículos, dependendo da velocidade e de condições climáticas. A princípio, a criação de sistemas que evitem este tipo de infração pode causar uma diminuição no prejuízo do condutor.

Os dados foram coletados através de filmagem, utilizando uma câmera fotográfica com resolução de 12 Mega pixels e dimensões do vídeo de 640px por 480px. O filmagem foi realizada com a câmera acoplada a uma motocicleta e guiada pela rodovia BA 148 (trecho de Livramento de Nossa Senhora-BA). A velocidade média da motocicleta nos vídeos utilizados como teste era de 45km/h. As imagens capturadas, figura 3.1, foram utilizadas na fase de rotulação das imagens e treinamento do classificador.

Uma das grandes dificuldades encontradas durante a coleta das imagens foi a trepidação sofrida pela motocicleta. Isto causava problemas no foco da imagem. A filmagem teve de ser refeita diversas vezes até a aquisição da melhor imagem possível.



Figura 3.1: Captura das imagens para treinamento do classificador
Fonte: Imagens adquiridas pelo autor

3.2 Rotular dados

Para rotular os dados foi utilizado o programa **positives builder**. Este programa foi implementado na linguagem C# e pode ser encontrado através do link <http://code.google.com/p/opencv-haar-cascade-positive-image-builder>.

Para cada imagem capturada ela gera uma linha de texto contendo o caminho do arquivo da imagem e as posições selecionadas do objeto. Abaixo temos nas figuras 3.2 e 3.3 um exemplo de captura da área onde contém a imagem de interesse e o respectivo dado gerado.

3.3 Extração de características, treinamento e validação

Após a rotulação dos dados partimos para a fase de treinamento do classificador. Nestas tarefas foram utilizados dois programas implementados pela biblioteca OpenCV:

- **opencv-createsamples** - este programa é responsável pela geração do vetor com cada rótulo associado a um objeto. Ele é utilizado para criar um arquivo de saída com exemplos positivos, figura 3.4. Ele extrai os recursos positivos das imagens antes de normalizar e redimensioná-los. Este aplicativo pode ser utilizado também para outras transformações na imagem como alteração de cor, transformações, adicionar ruídos, etc.

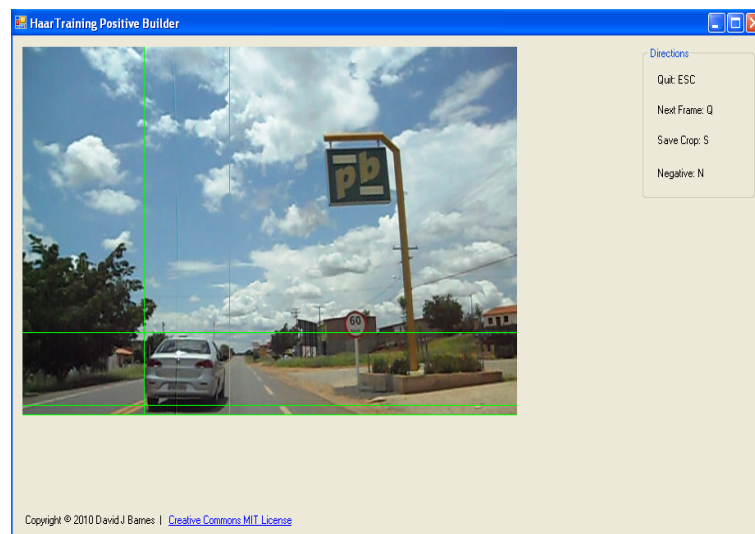


Figura 3.2: Rotular imagens utilizando o programa Positives Builder
Fonte: Imagens adquiridas pelo autor

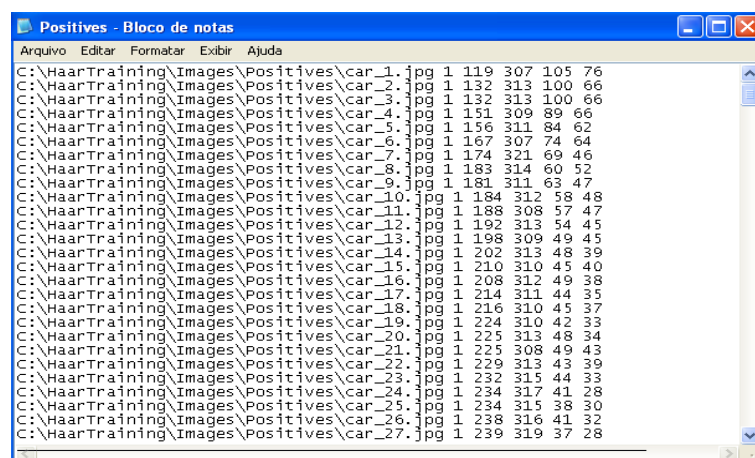


Figura 3.3: Dados rotulados após a escolha da região de interesse
Fonte: imagens adquiridas pelo autor

- **opencv-haartraining** - É o aplicativo criado com a implementação proposta por Viola e Jones (2001) e que foi utilizado na detecção dos veículos. Para o seu funcionamento é necessário informar a quantidade de imagens positivas e negativas coletadas, a taxa de erro na detecção dos dados, entre outros (figuras 3.5 e 3.6). Para esta fase os seguintes dados foram utilizados:

- Número de positivos: 400 imagens;
- Número de negativos: 390 imagens;
- Número de estágios para teste: 20;
- Taxa mínima para acerto: 0.998;
- Taxa máxima de falso alarme: 0.500

```

C:\Windows\system32\cmd.exe - haartraining.bat

C:\HaarTraining>createsamples.bat

C:\HaarTraining>opencv_createsamples.exe -info Positives.txt -vec Positives.vec
-num 400 -w 32 -h 24 PAUSE
Info file name: Positives.txt
Img file name: (NULL)
Vec file name: Positives.vec
BG file name: (NULL)
Num: 400
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 1.1
Max y angle: 1.1
Max z angle: 0.5
Show samples: FALSE
Width: 32
Height: 24
Create training samples from images collection...
Done. Created 400 samples

C:\HaarTraining>haartraining.bat

```

Figura 3.4: Extração de características
Fonte: Imagens adquiridas pelo autor

```

C:\Windows\system32\cmd.exe

C:\HaarTraining>opencv_haartraining.exe -data carros -vec Positives.vec -bg Negat
ives.vec -w 32 -h 24 -minhits 2 -minitrate 0.998 -maxfalsealarm 0.5 -npos 400
-nneg 390 -w 32 -h 24 -nonasm -mem 400 -mode ALL
Info file name: Positives.vec
Vec file name: Positives.vec
BG file name: Negatives.txt is a vecfile: no
Num pos: 400
Num neg: 390
Num stages: 20
Num splits: 2 (tree as weak classifier)
Mem: 400 MB
Symmetric: FALSE
Min hit rate: 0.998000
Max false alarm rate: 0.500000
Bg split training: 0.998000
Equal weights: FALSE
Width: 32
Height: 24
Applied boosting algorithm: GAB
Error (valid only for Discrete and Real AdaBoost): misclass
The number of splits in tree cascade: 0
Min number of positive samples per cluster: 500
Required test false alarm rate: 9.53674e-007

Tree Classifier
Stage
|
| 0 |
|---|

Number of features used : 460520
Parent node: NULL
*** 1 cluster ***
POS: 400 400 1.000000
NEG: 390
BACKGROUND PROCESSING TIME: 0.03
Recalculation time: 0.12

```

N	SMP	P	ST	THR	HR	FA	EXP	ERR

Figura 3.5: Início do treinamento do classificador
Fonte: Imagens adquiridas pelo autor

3.4 Testes

A próxima etapa após o treinamento do classificador foi a realização dos testes. Para isso foi utilizada a ferramenta Harpia que auxilia o programador na criação de rotinas para sistemas de visão computacional. Na primeira versão de testes foram capturadas 48 imagens positivas e 10 imagens negativas. O treinamento do classificador durou cerca de 10 minutos, porém a taxa de acerto não foi a desejada, conforme exibido na figura 3.7.

Na última tentativa foi realizada a captura de 400 imagens positivas e 390 imagens negativas. O treinamento desta vez teve duração de 22 horas e o resultado adquirido pode ser visualizado na figura 3.8. Como visto, a detecção dos objetos de interesse teve uma melhora significativa.

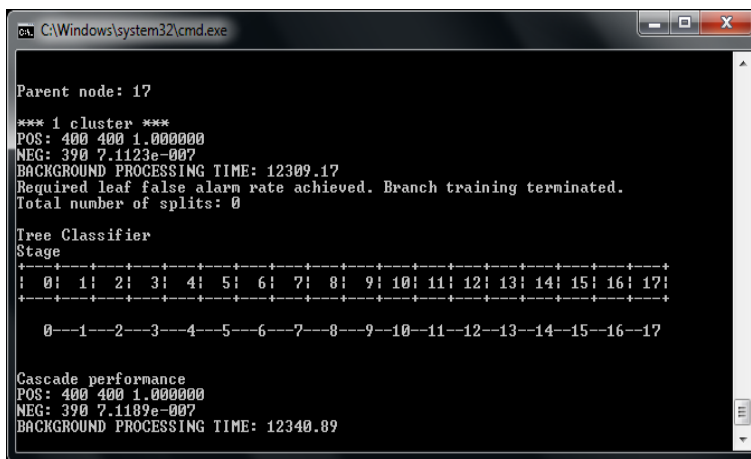


Figura 3.6: Término do treinamento do classificador
Fonte: Imagens adquiridas pelo autor

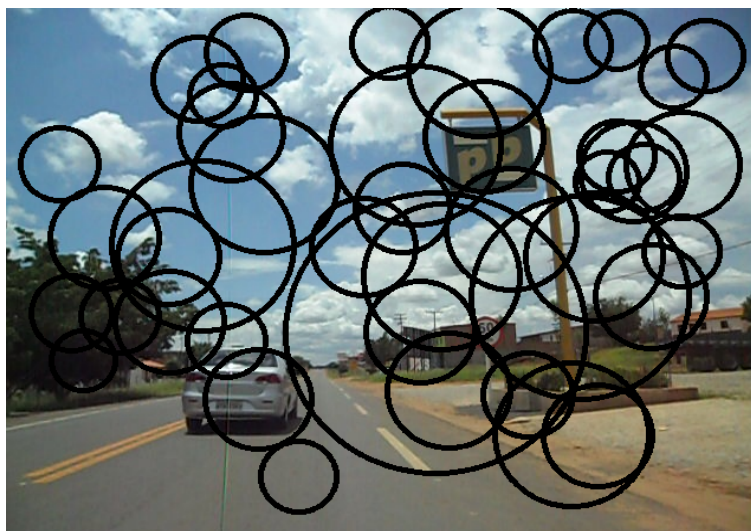


Figura 3.7: Primeira tentativa de detecção dos veículos
Fonte: Imagens adquiridas pelo autor

Quanto aos objetos identificados durante a execução do detector, é apresentada a tabela 3.1 contendo a taxa de detecção dos objetos nas imagens. São divididos em quatro grupos: Positivos, Negativos, Falsos Positivos e Falsos negativos. Os dados foram gerados após a realização do treinamento do classificador na segunda etapa.

Para confecção da tabela foram utilizadas 160 imagens positivas, com uma distância máxima do veículo de 40 metros.



Figura 3.8: Última tentativa de detecção dos veículos
Fonte: Imagens adquiridas pelo autor

Tabela 3.1: Taxa de detecção de veículos (%)

Detecção de veículos %		
	Positivos	Negativos
Positivos	78	4
Negativos	18	0

4 *Conclusões e trabalhos futuros*

4.1 **Conclusões**

Veículos são as melhores fontes de locomoção desde a sua criação até o presente momento. Várias adaptações já foram realizadas nele para melhoria da qualidade de direção e segurança do condutor. Ainda assim, como apresentado no trabalho, ainda existem muitos acidentes tendo como principal erro o "fator humano".

Muitos dispositivos podem ser criados para adaptação aos veículos e aumento da segurança do condutor. A visão computacional é um campo amplo de estudo. A idéia de fazer a máquina "enxergar" pode render diversos estudos e sistemas que auxiliam o homem no dia-a-dia em diversas áreas (medicina, geografia, história, economia, automobilismo, etc).

O aprendizado de máquina mostrou uma importante etapa na visão computacional, pois é através dela que os dados são armazenados na máquina para consulta posterior.

Durante a realização do projeto diversos problemas surgiram:

- Uma idéia do projeto era ser realizado utilizando somente software livre. Não foi possível realizar esta tarefa pois problemas surgiram na configuração das bibliotecas do OpenCV no sistema operacional Ubuntu.
- Outras duvidas surgiram quanto a resolução da câmera que iria capturar as imagens. Foi utilizada a melhor resolução da câmera (12 Megapixels). Porém, isto não descarta a realização de testes com uma câmera de resolução menor.
- De acordo as taxas de erros, acertos e quantidade de exemplares utilizados durante a fase de treinamento teve um bom resultado.

4.2 Trabalhos Futuros

A detecção de objetos foi realizada em apenas uma parte do veículo. Outros aspectos podem ser tratados e treinados como pedestres, placas, semáforos, e tantos outros dispositivos que ainda podem passar despercebido pelo homem.

Para trabalhos futuros pretende-se realizar aplicação de visão estéreo na detecção da distância dos veículos detectados. Isto será realizado após a aquisição de material apropriado para isto.

Outro projeto futuro é a realização de um comparativo entre os métodos de aprendizado de máquina: Redes Neurais, Classificadores, Árvores de decisão, Fuzzy, entre outras.

Referências Bibliográficas

- ADMINISTRATION, N. *The global positioning system: a shared national asset : recommendations for technical improvements and enhancements*. National Academy Press, 1995. ISBN 9780309052832. Disponível em: <<http://books.google.com/books?id=FAHk65slfY4C>>.
- AUTOMÁTICO, B. *Cambio Automatico*. 2010. Acesso em 24/12/2010. Disponível em: <<http://www.brasilautomatico.com.br/si/site/0303>>.
- AUTOPÉDIA. *AirBag*. 2010. Acesso em 30/11/2010. Disponível em: <<http://www.xl.pt/autopedia/seguranca/airbag.shtml>>.
- BARNESLEY, M. *Fractals everywhere*. [S.l.]: Academic Press, 1988.
- BRADSKY, G.; KAEHER, A. *Learning OpenCV*. [S.l.]: O'Reilly, 2008.
- BRASIL. *Novo Código de trânsito brasileiro*. Editora Auriverde, 2002. ISBN 9788586906046. Disponível em: <<http://books.google.com/books?id=GhpOAAAACAAJ>>.
- BURTON, D. et al. *Evaluation of Anti-lock Braking Systems Effectiveness*. [S.l.], 2004.
- CRUZEIROS, A. N. de. *GPS*. 2010. Acesso em 02/12/2010. Disponível em: <<http://www.ancruzeiros.pt/anci-gps.html>>.
- do frotista, B. do frotista (Ed.). *Freio ABS para todos os carros*. 2010. Acesso em 02/12/2010. Disponível em: <<http://www.blogdofrotista.com.br/freio-abs-para-todos-os-carros/>>.
- ECKERMAN, E. *World History of the automobile*. [S.l.]: Society of Automotive Engineers, Inc, 2001.
- ELSEVIER (Ed.). *ScienceDirect*. 2010. Acesso em 10/10/2010. Disponível em: <<http://www.sciencedirect.com/science/journal/02628856>>.
- ESPORTE, A. *Audi testa carro sem motorista*. 2010. Acesso em 14/12/2010. Disponível em: <<http://g1.globo.com/carros/noticia/2010/11/audi-testa-carro-sem-motorista.html>>.
- ESPORTE, A. *Carro viajou da Itália até a China sem motorista*. 2010. Acesso em 18/11/2010. Disponível em: <<http://g1.globo.com/tecnologia/noticia/2010/10/carro-viajou-da-italia-ate-china-sem-motorista.html>>.
- ESPORTE, A. (Ed.). *Google testa carro sem motorista nas ruas da Califórnia*. 2010. Acesso em 18/11/2010. Disponível em: <<http://g1.globo.com/tecnologia/noticia/2010/10/google-testa-carro-sem-motorista-nas-ruas-da-california.html>>.
- FORSYTH, D. A.; PONCE, J. *Computer Vision, a modern approach*. [S.l.]: Prentice Hall, 2002.

- FRERE, A. F.; DANGEL, R. J. Z.; CORDEIRO, E. C. Processamento de imagens para auxílio aos deficientes físicos. *II Congresso Latinoamericano de Ingeniería Biomédica*, 2001.
- GEOGRAPHIC, N. *Shock Absorber*. 2010. Acesso em 02/12/2010. Disponível em: <<http://news.nationalgeographic.com/news/2009/02/090219-shock-absorber-energy-cars.html>>.
- GEORGANO, G. *Early and Vintage Years 1886-1930: The Golden Era of Coach-building*. Mason Crest Publishers, 2003. (World of wheels). Disponível em: <<http://books.google.com/books?id=OeZVAAAACAAJ>>.
- GIGER, M. Computer-aided diagnosis of breast lesions in medical images. *Computing in Science & Engineering*, 2000.
- GIL, A. C. *Como elaborar projetos de pesquisa*. [S.l.]: Editora Atlas, 2002.
- GINIGE, A. A unified approach to image feature detection using finite state machine. *Image Processing and its applications*, 1992.
- HASTIE, T.; TIBISHIRANI, R.; FRIEDMAN, J. *The Elements of Statical Learning*. [S.l.]: Springer, 2009.
- HO, Y.; CHUA, C. 3d model building. *Image Processing and its applications*, 1999.
- HOLLANDA, J. B. de. Sinal verde para os carros elétricos. *Revista Ciência Hoje*, 2010.
- HYDRAUSYSTEM (Ed.). *DIRECAO HIDRAULICA*. 2010. Acesso em 24/12/2010. Disponível em: <<http://www.hidrausystem.com.br/>>.
- IDA, T. Image segmentation and contour detection using fractal coding. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, v. 8, 1998.
- INIGO, R. Application of machine vision to traffic monitoring and control. *IEEE Transactions on Vehicular Technology*, 1989.
- JIN, X.; DAVIS, C. H. Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks. *Image and Vision Computing*, 2007.
- JIN, X. et al. Detection of small lesions in medical x-ray image based on multi-scale fractal feature. *IEEE*, 2009.
- KINSNER, W. A unified approach to fractal dimensions. *Intern. Conf. Cognitive Informatics*, 2005.
- KONIG, A.; GENTHER, H.; GLESNER, M. A generic dynamic inspection system for visual object inspection and industrial quality control. *International Joint Conference on Neural Networks*, 1993.
- KUSANO, A. et al. 3d inspection of electronic devices by means of stereo method on single camera environment. *IEEE SMC*, 2008.
- LENSKY, A. A.; JO, K.-H.; GUBAREV, V. V. Vehicle license plate detection using local fractal dimension and morphological analysis. *IEEE*, out. 2006.

- LIU, Y.; PAYEUR, P. Vision-based detection of activity for traffic control. *IEEE CCECE*, 2003.
- LO, S.-C. B. et al. Artificial convolution neural network techniques and applications for lung nodule detection. *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 1995.
- LUKAC, R.; PLATANIOTIS, K. N. (Ed.). *Color Image Processing - Methods and Applications*. [S.l.]: CRC Press, 2007.
- MEIR, R.; RÄTSCH, G. An introduction to boosting and leveraging. *Advanced lectures on machine learning*, 2003. ISSN 3-540-00529-3.
- MILLER, F.; VANDOME, A.; MCBREWSTER, J. *Automatic Transmission*. VDM Publishing House Ltd., 2009. ISBN 9786130226916. Disponível em: <<http://books.google.com/books?id=ILjgQgAACAAJ>>.
- MOUNTNEY, P.; STOYANOV, D.; YANG, G.-Z. Three dimensional tissue deformation recovery and tracking. *IEEE SIGNAL PROCESSING MAGAZINE*, 2010.
- MUYUN, W.; MINGYI, H. Image feature detection and matching based on susan method. *ICICIC*, 2006.
- NUNES, F. L. S. Introdução ao processamento de imagens médicas para auxílio ao diagnóstico – uma visão prática. In: _____. [S.l.]: Sociedade Brasileira de Computação, 2006. p. 73–126.
- NUNNEY, M. *Light and heavy vehicle technology*. Butterworth-Heinemann, 2006. Disponível em: <<http://books.google.com/books?id=eL6TBaSnD78C>>.
- S2I. *S2i*. 2010. Acesso em 10/10/2010. Disponível em: <<http://s2i.das.ufsc.br>>.
- SCHAPIRE, R. E. The boosting approach to machine learning an overview. *Nonlinear Estimation and Classification*, 2003.
- SCHNEIDERMAN, H.; KANADE, T. A statistical method for 3d object detection applied to faces and cars. *IEEE CONFERENCES*, 2000.
- SEBE, N. et al. *Machine Learning in Computer Vision*. [S.l.]: Springer, 2005.
- SHAO, J. Global image feature correspondence under a multi-image network. *Image and Vision Computing*, 1999.
- STOKMAN, H.; GEVERS, T. Selection and fusion of color models for image feature detection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 2007.
- THOMPSON, W. et al. Feature-based reverse engineering of mechanical parts. *IEEE TRANSACTIONS ROBOTIC AND AUTOMATION*, 1999.
- TRUCCO, E.; VERRI, A. *Introductory Techniques for 3D Computer Vision*. [S.l.]: Prentice Hall, 1998.
- UCHIKAWA, K. Color vision. *IEEE SMC*, 1999.
- UNESP (Ed.). *Acidentes no transito*. 2010. Acesso em 24/12/2010. Disponível em: <<http://www.bauru.unesp.br/cursocipa/artigos/4transito.htm>>.

UNIÃO, R. *Amortecedor*. 2010. Acesso em 30/11/2010. Disponível em:
<<http://www.recuperadorauniao.com.br/>>.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. *International Journal of Computer Vision*, 2001.

WIKIPEDIA. *Computer vision*. 2010. Acesso em 18/11/2010. Disponível em:
<<http://en.wikipedia.org/wiki/ComputerVision>>.

WIKIPEDIA. *Fractal*. 2010. Acesso em 18/11/2010. Disponível em:
<<http://pt.wikipedia.org/wiki/Fractal>>.

WIKIPEDIA. *LaTeX*. 2010. Acesso em 30/11/2010. Disponível em:
<<http://pt.wikipedia.org/wiki/LaTeX>>.

WIKIPEDIA. *Photogrammetry*. 2010. Acesso em 01/11/2010. Disponível em:
<<http://en.wikipedia.org/wiki/Photogrammetry>>.

XIE, S. et al. Detection and tracking of moving object in visual tracking from a low-altitude flying helicoptershaorong xie. *International Conference on Robotics and Biomimetics*, p. 15–18, 2007.

XPLORE, I. (Ed.). *IEEE Xplore*. 2010. Acesso em 30/11/2010. Disponível em:
<<http://ieeexplore.ieee.org/Xplore/guesthome.jsp?reload=true>>.

YAP, M. H.; EDIRISINGHE, E. A.; BEZ, H. E. Object boundary detection in ultrasound images. *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision*, 2006.

ZHOU, Q.; MA, L.; CHELBERG, D. Adaptive object detection and recognition based on a feedback strategy. *Image and Vision Computing*, v. 24, 2006.

APÊNDICE A – Código fonte gerado pela ferramenta Harpia

Listing A.1: Código fonte detector

```

// Auto-generated C Code – S2i Harpia
/*
 *      In order to compile this source code run, in a terminal window, the
 *      following command:
 *      gcc sourceCodeName.c 'pkg-config --libs --cflags opencv' -o
 *      outputProgramName
5 *
 *      the 'pkg-config ... opencv' parameter is a inline command that
 *      returns the path to both
 *      the libraries and the headers necessary when using opencv. The
 *      command also returns other necessary compiler options.
 */
// header:
10 #include <stdio.h>
#include <stdlib.h>
#include <cv.h>
#include <cvaux.h>
#include <highgui.h>
15 #include <math.h>

#define PI 3.1415926535898
double rads(double degs)
{
20     return (PI/180 * degs);
}

int main(int argc, char ** argv)
{
25     //declaration block
IplImage * block2_img_01 = NULL;

```

```

IplImage * block1_img_i1 = NULL;
CvPoint block1_point_o1 = cvPoint(0,0);
CvRect block1_rect_o2 = cvRect( 0, 0, 1, 1);
30 IplImage * block1_img_o3 = NULL;
double block1_double_o4 = 0.0;
static CvMemStorage* block1_storage = 0;
static CvHaarClassifierCascade* block1_cascade = 0;
const char* block1_cascade_name = "/home/jadson/Documentos/UESB/Monografia/
    HaarTraining_Windows7/carros.xml";
35 double block41_double_i1;
double block41_double_o1;
IplImage * block39_img_i1 = NULL;
IplImage * block39_img_o1 = NULL;
int end; end = 0; int key;
40 CvCapture * block2_capture = NULL;
    IplImage * block2_frame = NULL;
    block2_capture = cvCreateFileCapture("/home/jadson/Documentos/UESB/
        Monografia/Videos/video_teste2");
while (!end)
    {
45 cvGrabFrame (block2_capture);
        block2_frame = cvRetrieveFrame (block2_capture);

        //execution block
        //Weight: 1
50 // Video Mode
        block2_img_o1 = cvCloneImage( block2_frame );
        block1_img_i1 = cvCloneImage(block2_img_o1); // IMAGE conection
        //Weight: 2

55 if(block1_img_i1){
            double scale = 1.3;
            block1_cascade = (CvHaarClassifierCascade*)cvLoad(
                block1_cascade_name, 0, 0, 0 );
            IplImage* gray = cvCreateImage( cvSize(block1_img_i1->width,
                block1_img_i1->height), 8, 1 );
            IplImage* small_img = cvCreateImage( cvSize( cvRound (block1_img_i1
                ->width/scale), cvRound (block1_img_i1->height/scale)),8, 1 );
60 cvCvtColor( block1_img_i1, gray, CV_BGR2GRAY );
            cvResize( gray, small_img, CV_INTER_LINEAR );
            cvEqualizeHist( small_img, small_img );
            if (!block1_img_o3)
                block1_img_o3 = cvCloneImage(block1_img_i1);

```

```

65     cvCopy( block1_img_i1 , block1_img_o3 , 0 );
        block1_storage = cvCreateMemStorage( 0 );
        cvClearMemStorage( block1_storage );
        block1_rect_o2 = cvRect( 0, 0, 1, 1 );
        CvSeq* faces = cvHaarDetectObjects( small_img , block1_cascade ,
            block1_storage , 1.1 , 2.0 , 0/*CV_HAAR_DO_CANNY_PRUNING*/ , cvSize
            (30, 30) );
70     block1_double_o4 = faces->total;
        if( faces )
        {
            int i;
            for( i = 0; i < ( faces ? faces->total : 0); i++ )
75         {
            CvRect* r = (CvRect*)cvGetSeqElem( faces , i );
                if( r )
                {
                    CvPoint center;
80                 int radius;
                    center.x = cvRound((r->x + r->width*0.5)*
                        scale);
                    center.y = cvRound((r->y + r->height*0.5)*
                        scale);
                    radius = cvRound((r->width + r->height)
                        *0.25*scale);
                    cvCircle( block1_img_o3 , center , radius ,
                        cvScalarAll(0) , 3, 8, 0 );
85                 if( i == 0 )
                    {
                        block1_point_o1 = center;
                        block1_rect_o2.x = (r->x)*scale;
                        block1_rect_o2.y = (r->y)*scale;
90                     block1_rect_o2.width = (r->width)*
                        scale;
                        block1_rect_o2.height = (r->height)
                            *scale;
                    }
                }
            }
        }
95     }
        cvReleaseImage( &gray );
        cvReleaseImage( &small_img );
    }
    block39_img_i1 = cvCloneImage(block1_img_o3); // IMAGE conection

```

```
100 block41_double_i1 = block1_double_o4; // DOUBLE conection
    //Weight: 3

    if(1){
    char outPutStr[102];
105 snprintf(outPutStr,102,"export HRP_DB=%f;echo $HRP_DB >> /home/jadson/
        Documentos/UESB/Monografia/video_teste2.txt" ,(float)block41_double_i1);
        system(outPutStr);}
    block41_double_o1 = block41_double_i1;
    //Weight: 3

    if(block39_img_i1){
110 block39_img_o1 = cvCloneImage(block39_img_i1);
    cvNamedWindow("block39_img_o1",CV_WINDOW_AUTOSIZE );
    cvShowImage("block39_img_o1",block39_img_i1);}

        key = cvWaitKey (33);
115 if(key != -1)
    end = 1;

        //deallocation block
    cvReleaseImage(&block2_img_o1);
    cvReleaseImage(&block1_img_o3);
120 cvReleaseImage(&block1_img_i1);
    cvReleaseMemStorage(&block1_storage);
    //nothing to deallocate
    cvReleaseImage(&block39_img_o1);
    cvReleaseImage(&block39_img_i1);
125 }
    cvReleaseCapture(&block2_capture);
    return 0;
    } //closing main()
```
