



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Trabalho de Conclusão de Curso

Michel de Souza Pinto

Av. Getúlio Vargas, 1200 – Tel. (XX18)3302-1055 – CEP 19.807-634 – ASSIS –SP

home page: www.fema.edu.br - e-mail: imesa@femanet.com.br



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

Identificador de Placas de Carro

ALUNO: Michel de Souza Pinto

ORIENTADOR: Regina Fumie Eto

ASSIS – SP

2009

Identificador de Placas de Carro

Michel de Souza Pinto

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação em Bacharelado em Ciência da Computação analisado pela seguinte comissão examinadora:

Profª Regina Fumie Eto
(orientador) IMESA – FEMA

Prof.
(orientador) IMESA – FEMA

Prof.
(orientador) IMESA – FEMA

“Dedico este Trabalho a todos que depositaram suas confianças, principalmente a minha orientadora que acreditou, soube entender e dedicar – se ao projeto. Agradeço a toda minha família, principalmente minha mãe e meu pai, por passar todo ensino de vida. Dedico a minha namorada por compreensão e apoio extra. Mas esses longos dias que se passaram, jamais poderei me esquecer do meu irmão que me apoiou desde às primeiras horas, com sua maneira didática e intelectual, apoiando e reservando seu tempo á ajudar, obrigado”

“...A memória de meu grande amigo Carlos Henrique Ferreira Spera (Charlie) ...”

“A coisa mais bela que podemos experimentar é o mistério. Essa é a fonte de toda a arte e ciências verdadeiras”

Albert Einstein

AGRADECIMENTOS

Agradeço primeiramente a DEUS, ele é o responsável por todos os meus méritos, meu mestre, minha razão de viver.

Aos meus pais que apóiam tudo na minha vida, e amo demais.

Aos professores Ligia Batista (UTFPR), Maurício Lemes (UTFPR), Felipe Manfio e Elcio (IME), que por intermédio do professor Thiago Pinto (UTFPR) contribuíram com a realização deste trabalho.

Por fim, agradeço todas as pessoas que ajudaram de maneira formal ou informal, obrigado pela paciência e entendimento.

RESUMO

O objetivo deste trabalho é o desenvolvimento de um sistema de reconhecimento de caracteres alfanuméricos utilizando redes neurais aplicado na identificação de placas de carros para automatização do controle de veículos na portaria da empresa Máquinas Agrícolas Jacto S.A, localizada na cidade de Pompéia – SP.

Este trabalho apresenta um estudo das técnicas de processamento de imagens e de visão computacional, encontradas na literatura existente, necessárias para o desenvolvimento do sistema. Basicamente o sistema, a partir de uma imagem digital do carro contendo a placa, localiza a região da imagem contendo a placa, faz a extração dos caracteres e o reconhecimento dos mesmos.

A linguagem de programação escolhida para a implementação do sistema é a Java com ambiente de desenvolvimento Netbeans. Para a construção do banco de dados é utilizado o sql gratuito Firebird.

Palavra Chave: Reconhecimento de Caracteres, Processamento de Imagem, Redes neurais.

ABSTRACT

This study is the development of a system of recognition of alphanumeric characters using neural networks applied in identifying license plates for automated control of vehicles in the lobby of the company of the company JACTO S.A Agricultural Machinery, located in the city of Pompéia - SP.

This paper presents a study of the techniques of image processing and computer vision, found in the literature, for the development of the system. Basically the system, from a digital image of the car containing the card, locate the image region containing the card, is the extraction of characters and recognition.

The programming language chosen for the implementation of the system is a Java development environment on Netbeans. For the construction of the database is used the sql free Firebird.

Key words: Character Recognition, Image Processing, Neural networks.

SUMÁRIO

Lista de Figuras	1
Lista de Fórmulas	3
Lista de Tabelas	4
Lista de Símbolos	5
1 - Introdução	6
1.1 Considerações Iniciais	6
1.2 Motivação	7
1.3 Objetivos	7
1.4 Justificativa e Metodologia	7
1.5 Problemática	8
1.6 Organização do Trabalho	9
2 - Conceitos Fundamentais	11
2.1 Introdução	11
2.2 Imagem	11
2.2.1 Imagem Digital.....	11
2.2.2 Imagem Binária	12
2.2.3 Imagem Colorida	12
2.2.4 Aquisição Imagem	13
2.2.5 Formatos	13
2.2.6 Cores	14
2.3 Processamento de Imagens	15
2.3.1 Transformação de uma Imagem em Binária.....	15
2.3.2 Realce	16
2.3.3 Definição do Histograma de uma Imagem	17
2.3.4 Bordas	19
2.3.5 Limiarização	19
2.4 Elementos de um Sistema de Processamento digital de imagens	22
2.5 Análise de Imagem	23

3 -	Sistemas de Reconhecimento	24
3.1	Introdução	24
3.2	Localização da Placa..	25
3.3	Processos de Verificação da Placa e Extração de Caracteres	27
3.4	Técnicas de Reconhecimento de Caracteres.....	30
3.4.1	Redes Neurais Artificiais	30
4 -	Implementação do Sistema Proposto	36
4.1	Introdução	36
4.2	Java	36
4.3	Firebird	37
4.4	Estruturas da Implementação	37
4.4.1	Pacote j_reconhecer_placa	39
4.4.2	Pacote bd_banco_dados	44
4.4.3	Interface do Sistema	45
5 -	Conclusões e Perspectivas Futuras	47
	Referências Bibliográficas	48

LISTA DE FIGURAS

Figura 1.1 - Identificação do Carro e sua Liberação na Portaria de Veículos	8
Figura 1.2 - Instalações da empresa Jacto S.A	9
Figura 2.1 - Formação da Imagem Colorida	12
Figura 2.2 - Exemplo de Imagem com Variação do Limiar	16
Figura 2.3 - Exemplo de Histograma	17
Figura 2.4 - Imagem Digital e seu Histograma	17
Figura 2.5 - Imagem Escura e seu Histograma	18
Figura 2.6 - Imagem com bom contraste e seu Histograma	18
Figura 2.7 - (a) Imagem Original, (b) Imagem Aplicado Filtro Sobel	19
Figura 2.8 - Imagem e seu Histograma	20
Figura 2.9 - Imagem Segmentada e escolha do Limiar Global	21
Figura 2.10 - (a) Imagem, (b) Histograma	21
Figura 2.11 - Imagem Segmentada e escolha do limiar Global	22
Figura 3.1 - Estrutura do Sistema de Reconhecimento de Placa	24
Figura 3.2 - Região candidata encontrada durante a Localização	25
Figura 3.3 - Corte Transversal da Imagem do Carro	26
Figura 3.4 - (a) Imagem Original, (b) Variação Tonal das Linhas A e B	27
Figura 3.5 - (a) Imagem Original, (b) Imagem da Placa Encontrada	28
Figura 3.6 - Projeção Horizontal e Vertical	28
Figura 3.7 - Segmentação dos Caracteres	29
Figura 3.8 - Estágios Utilizados na Busca Por Padrões	30
Figura 3.9 - Modelo Geral de um Neurônio Artificial proposto por Mcculloch <i>et al.</i> ...	31
Figura 3.10 - Sistema Proposto para Reconhecimento de Veículos	32
Figura 3.11 - Doze Códigos de Vetores obtidos por 12 linhas Horizontais	33
Figura 3.12 - Exemplo de Detecção de Ruídos	33
Figura 3.13 - Tabela de Códigos utilizada para Codificação de Caracteres da Placa	34
Figura 3.14 - Caractere 7 Obtido de Quatro Placas Diferentes	35
Figura 4.1 - Modelo de Caso de Uso Geral	37
Figura 4.2 - Modelo de Caso de Aprendizagem	38

Figura 4.3 - Modelo de Caso de Reconhecimento	38
Figura 4.4 - Modelo de Classe	39
Figura 4.5 - Código Fonte da Classe aut_Placa	40
Figura 4.6 - Algoritmo de Sobel	41
Figura 4.7 - Imagens	42
Figura 4.8 - Caracteres Encontrados Pelo Sistema	43
Figura 4.9 - Exemplo da Tabela códigos_caracteres	44
Figura 4.10 - Código Fonte da classe conexão	45
Figura 4.11 - Interface do Sistema	45
Figura 4.12 - Execução do Sistema Proposto	46

LISTA DE FÓRMULAS

Fórmula 2.1 - Transformação da Imagem em Tons de Cinza	14
Fórmula 2.2 - Limiarização Simples	20

LISTA DE TABELAS

Tabela 3.1 - Codificação da figura 3.14	35
---	----

LISTA DE SÍMBOLOS

BMP -	<i>Business Process Modeling</i>
CCD -	<i>Charge Coupled Device</i>
CONTRAN -	Código de Trânsito Brasileiro
DDR2 -	<i>Double Data Rate</i>
DETRAN -	Departamento Estadual de Trânsito
GB -	<i>Giga Byte</i>
HP -	<i>Hewlett Packard</i>
JPEG -	<i>Joint Photographic Experts Group</i>
S.A -	Sociedade Anônima
TDC -	Transformada Direta do Cosseno
T.I -	Tecnologia da Informação
2X -	Processador com 2 núcleos

1 Introdução

1.1 Considerações Iniciais

Neste mundo competitivo e dinâmico, a informática vem se destacando devido ao crescimento que a mesma pode proporcionar aos mais diversos setores existentes. Atualmente, as empresas vêem com bons olhos os trabalhos que pertencem à área tecnológica, buscando melhorias que podem torná-la mais competitiva no mercado. Nota-se que o Brasil vem progredindo muito em todas as áreas, inclusive no desenvolvimento de softwares gerenciais e de monitoramento.

A visão é considerada um dos principais meios de percepção do mundo exterior pelos seres humanos. Para o auxílio de tarefas que utilizam a visão como função principal podem ser empregados sistemas de visão artificial capazes de reconhecer padrões previamente ensinados em uma imagem complexa. A leitura automática é considerada uma das mais atraentes tarefas nessa área (Souza, 2000). O principal objetivo de um sistema de visão artificial é adquirir, processar, analisar e interpretar um determinado tipo de imagem para uma determinada finalidade.

(Gonzalez *et al.*, 1993) divide o processamento de imagem digital em duas principais aplicações: melhora do aspecto visual de uma imagem para interpretação humana, e processamento dos dados de uma cena para percepção automática de máquinas.

Segundo (Campos *et al.*, 2001) o desempenho de um sistema de visão artificial é um item fundamental, por isso são desenvolvidos para aplicações bem específicas, permitindo que o sistema apresente um alto desempenho para uma determinada classe de imagem.

Com o avanço contínuo da tecnologia e a diminuição dos custos relativos aos computadores e periféricos, a análise de imagem digital está numa crescente expansão, sendo usada em diversas áreas: robótica, automação, astronomia, biologia, agricultura, entre outros (Souza, 2000).

1.2 Motivação

A motivação da realização desse trabalho vem da possibilidade de construir um trabalho acadêmico com potencial de aplicação no meu cotidiano, visto que atualmente trabalho no setor de tecnologia da empresa Jacto S.A, e verifico a real necessidade de automatizar o processo de identificação de carros na portaria de veículos dessa empresa.

1.3 Objetivos

O objetivo deste trabalho é dividido em duas fases, onde a primeira consiste desenvolver uma linha de pesquisa relacionada ao reconhecimento da imagem, obtendo materiais e ferramentas que demonstram como localizar a placa do veículo e identificar seus caracteres.

A segunda fase é a construção do software na linguagem Java. Para construir e finalizar a estrutura do trabalho deve-se basear o ambiente de tratamento utilizando um conjunto de conhecimentos interdisciplinares. Esta previsibilidade poderá gerar retornos futuros, se bem aplicado, um controle mais efetivo sobre veículos na portaria.

1.4 Justificativas e Metodologia

Como atualmente trabalho na área de T. I (Tecnologia da Informação) da empresa Máquinas Agrícola Jacto S.A possuo acesso aos projetos futuros que a mesma coloca para a modernização de todos os setores. Entre esses projetos, existe um cujo foco é a automatização da portaria de entrada de carros, pensando neste projeto construo um sistema que visa aumentar a facilidade no fluxo de carros que adentram á empresa.

Para o desenvolvimento desse trabalho inicialmente é realizado um estudo de conceitos relacionados ao processamento de imagem e de algumas técnicas destinadas ao reconhecimento de caracteres alfanuméricos. Em seguida, o sistema é desenvolvido na plataforma Windows XP Professional, utilizando para testes o notebook HP DV600, com capacidade 2 gb de memória ddr2 e processador Turion 64 2x 1.8 mhz. O ambiente de desenvolvimento é o Netbeans 6.0.1, banco de dados Firebird 1.5 e a tecnologia escolhida foi Java, sendo que o escopo do trabalho se restringe ás etapas de localização e reconhecimento de caracteres de uma placa de carro.

1.5 Problemática

O projeto será aplicado na empresa Máquinas Agrícola Jacto S.A, localizada em Pompéia – SP. Fundada em 1948 pelo Sr Shunji Nishimura, empresa multinacional voltada para o agricultor, tendo como principal negócio, pulverizadores e colhedeiças de café.

Apesar da empresa Jacto S.A destacar-se no cenário nacional e internacional, investindo sempre em tecnologias inovadoras, a mesma enquadra-se entre as empresas que ainda não possuem identificação tecnológica de imagens em suas portarias, principalmente na portaria de veículos.

A identificação e liberação do carro na portaria de veículos da empresa são realizadas manualmente pelos seguranças (figura 1.1), o motorista é abordado, e em seguida, após a sua identificação é liberado para adentrar as instalações da empresa (figura 1.2).



Figura 1.1. Identificação do carro e sua liberação na portaria de veículos da empresa.



Figura 1.2. Instalações da empresa Jacto S.A.

1.6 Organização do Trabalho

O presente trabalho está organizado na seguinte estrutura:

- Capítulo 1: Descreve a introdução inicial do trabalho, passando para o leitor informações gerais tais como: objetivos, justificativas, metodologia e problemática.
- Capítulo 2: São apresentados os detalhes sobre processamento de imagem, conceitos fundamentais sobre a imagem, aquisição da mesma, formatos, elementos da imagem, histogramas, bem como técnicas necessárias para o realce e filtragem da imagem.
- Capítulo 3: O capítulo apresenta etapas do sistema de reconhecimento de placa de carro. É apresentada uma breve revisão dos textos que possuem alguma influência sobre o sistema desenvolvido nesse trabalho.

- Capítulo 4: O sistema de reconhecimento de placa desenvolvido neste trabalho junto com algumas partes do seu código fonte. Comentam-se os resultados obtidos.
- Capítulo 5: Conclusões e considerações para investigações futuras são apresentadas neste capítulo.

2 Conceitos Fundamentais

2.1 Introdução

Neste capítulo serão revisados alguns conceitos fundamentais para construção do sistema. Também serão demonstradas algumas técnicas de segmentação de imagens, que auxiliarão no desenvolvimento do sistema

2.2 Imagem

Segundo (Manfio, 2008) a imagem é uma representação gráfica de pessoa ou objeto, formada pelo conjunto de pontos no espaço, para onde convergem, ou de onde divergem, os raios luminosos que, originados de um objeto luminoso ou iluminados, passam através de um sistema óptico. Para o ser humano uma imagem é o que ele percebe como resultado da luz atingindo seus olhos, passando pelo cristalino e incidindo na retina.

2.2.1 Imagem Digital

Como os computadores não possuem a capacidade de manipular dados analógicos é necessário converter a imagem contínua em um formato digital. O termo imagem digital refere-se a uma imagem que pode ser discretizada quanto as suas coordenadas espaciais e quanto á intensidade de seu brilho (Gonzalez *et al.*, 1993).

De acordo com (Bellon, 1996) e (Souza, 2000), a imagem digital é uma função bidimensional $f(x,y)$, onde f é o valor da intensidade do brilho da imagem para um ponto qualquer (x,y) pertencente á imagem. As coordenadas (x,y) são utilizadas para localizar um ponto qualquer pertencente a imagem em questão.

A imagem digital possui algumas características fundamentais, como a resolução e o número de cores, e podendo ser representada, processada e armazenada no formato de uma imagem matricial ou vetorial. A imagem matricial é formada por uma série ordenada de pixels dispostos continuamente em linha e colunas, chamada de matriz de pontos, onde cada pixel tem suas características definidas com cor e brilho. A imagem vetorial é formada por um conjunto de instruções matemáticas, onde ela é gerada a partir das instruções que são processadas, neste modelo pode-se alterar o tamanho sem que haja perda de definição (Manfio, 2008).

2.2.2 Imagem Binária

Denomina-se imagem binária aquela que contém dois níveis de cores, sendo eles o preto e o branco. Cada pixel de uma imagem binária necessita de 1 bit para ser representada, a analogia é 0 (para preto) e 1 (para branco).

2.2.3 Imagem Colorida

A retina do olho humano é coberta por células foto-receptoras que são funcionalmente análogas aos receptores em um chip de CCD. Estas células absorvem a luz de uma imagem que é focada sobre a retina pelas lentes e córneas, e geram impulsos nervosos que viajam para o cérebro humano, via nervo óptico. A frequência destes impulsos codifica o brilho da luz incidente (Souza, 2000).

Uma imagem digital colorida é vista como a sobreposição de três imagens monocromáticas, cada uma delas representando a intensidades da cores vermelho (red - R), verde (green - G) e azul (blue - B) que compõem a imagem original (Souza, 2000). Sendo assim, o pixel na imagem colorida é o resultado da sobreposição das intensidades dos pixels espacialmente alinhados das imagens monocromáticas (figura 2.1). Geralmente, os valores dos pixels das imagens correspondem a cada uma das três cores primárias, vermelho, verde e azul e são representadas por números inteiros entre 0 e 255, assim sendo, são necessários 24 bits para sua representação, 8 bits para cada uma dos três componentes.

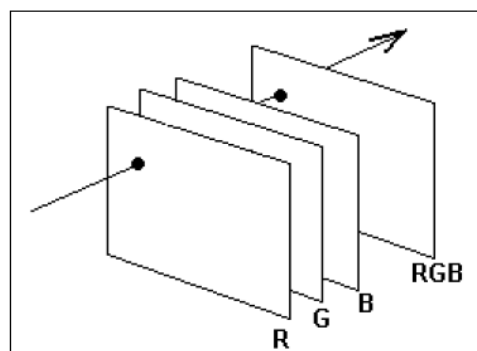


Figura 2.1. Formação da imagem colorida (Souza, 2000)

2.2.4 Aquisição de Imagem

A transformação de uma imagem em uma imagem digital consiste no processo de aquisição de imagem. Entre os dispositivos existentes destinados a aquisição de imagens, citamos: câmera digital, scanner, câmera de vídeo.

Neste trabalho o dispositivo para aquisição de imagem utilizada será a câmera digital. A câmera possui uma matriz de células semicondutoras fotossensíveis, que atuam como capacitores, armazenando carga elétrica proporcional à energia luminosa incidente. O sinal elétrico produzido é condicionado por circuitos eletrônicos especializados, produzindo à saída um sinal composto de vídeo analógico (Marques *et al.*, 1999). Este sinal obtido à saída do dispositivo, deve ser submetido à discretização espacial e em amplitude para adquirir um formato computacional.

2.2.5 Formatos

Existem muitos formatos de armazenamento disponíveis pelos softwares que manipulam imagens digitais, cita-se como exemplo os arquivos com extensões: JPEG (*Joint Photographic Experts Group*) e o BMP (*Windows Bitmap*).

O JPEG é um método de compressão para imagens fotográficas, e para comprimir a imagem são usados cinco métodos.

- Conversão da Imagem
- Sub-Montagem
- TDC – Transformada Discreta Do Cosseno
- Quantização
- Codificação da Entropia

BMP significa mapa de bits, este arquivo é uma representação fiel das informações contidas na imagem sem qualquer tipo de compressão.

2.2.6 Cores

Os modelos de cores são responsáveis pelas traduções de cores em dados numéricos consistentes para serem usados na computação gráfica. Abaixo verificamos os modelos usados.

- RGB
- CYMK
- TONS DE CINZA

RGB : as cores são representadas pela adição das cores primária vermelha (R), verde (G) e azul (B) em várias proporções e intensidade, é utilizado para trabalhar com imagens digitais, pois se baseiam nas mesmas propriedades fundamentais da luz que ocorre na natureza.

Cada cor primária tem um intervalo de 0 a 255. O número total de cores que poderão ser formadas é de aproximadamente 16,7 milhões de cores e pode se formar 256 possibilidades de tons de cinza. Geralmente ele é utilizado em monitores de computadores e televisão, pois a ausência de cores forma a cor preta e os controles primários em seu valor máximo, 255, formam o branco. Nesse trabalho todos os exemplos utilizaram o modelo RGB.

CYMK : Baseia-se na subtração de cores, onde o C é o ciano, o M é o magenta, o Y é o amarelo e o K é o preto. Utiliza-se K para não se confundir com o azul (*blue*). Como esse modelo se baseia na subtração de cores é utilizado nas impressoras, pois a ausência de cor forma o branco.

TONS DE CINZA: baseada no modelo RGB, contém os três controles primários, vermelho (R), verde (G) e azul (B), com o mesmo valor. A transformação de uma imagem colorida no modelo RGB para uma imagem em tons de cinza é feita utilizando a fórmula,

$$X = \frac{R + G + B}{3},$$

Fórmula 2.1. Transformação de imagem em tons cinza.

onde X será o novo valor dos controle primários R, G e B.

2.3 Processamento de Imagens

O estudo de reconhecimento de caracteres alfanuméricos para identificação de placas de veículos, através do processamento de imagem, tem início na década de 70, quando foram desenvolvidos os primeiros protótipos (Polidório *et al.*, 1997).

Segundo (Gonzalez *et al.*, 1993), o processamento de imagens é um dos passos necessários de um sistema de processamento digital, tem como objetivo realizar alterações na imagem original, melhorar a visualização e destacando partes.

Entre as técnicas existentes citamos:

- Imagem Binária
- Realce
- Equalização por Histograma
- Identificar Borda

SONKA *et al.* (2007), afirma a existência de 2 níveis de processamento de imagens, baixo e alto. O baixo refere-se á aquisição de imagens e as opções de pré-processamento da imagem (ajuste de histograma, filtragem para remoção de ruídos), enquanto o nível alto consiste em tarefas de reconhecimento e interpretação.

Ainda, segundo (SONKA *et al.*, 2007) o processo de reconhecimento consiste na classificação das regiões da imagem para um fim específico, métodos estatísticos, sintáticos e conexionistas são empregados neste processo. O processo de interpretação entende o contexto geral apresentado na imagem, os métodos que costumam ser empregados são a lógica formal, redes semânticas e sistemas de produção.

2.3.1 Transformação de uma Imagem em Binária

Para transformar uma imagem em uma imagem binária deve-se primeiramente transformá-la em uma imagem em tons de cinza, depois se deve definir um limiar. O limiar é um valor arbitrário, onde todos os valores de tons de cinzas da imagem que estão abaixo do limiar serão transformados em 0 (zero), e todos acima do limiar serão transformados em 1 (um). Pode-se ver na figura 2.2 a diferença da imagem binária variando o limiar, quanto maior o limiar maior será a área da imagem visualizada.

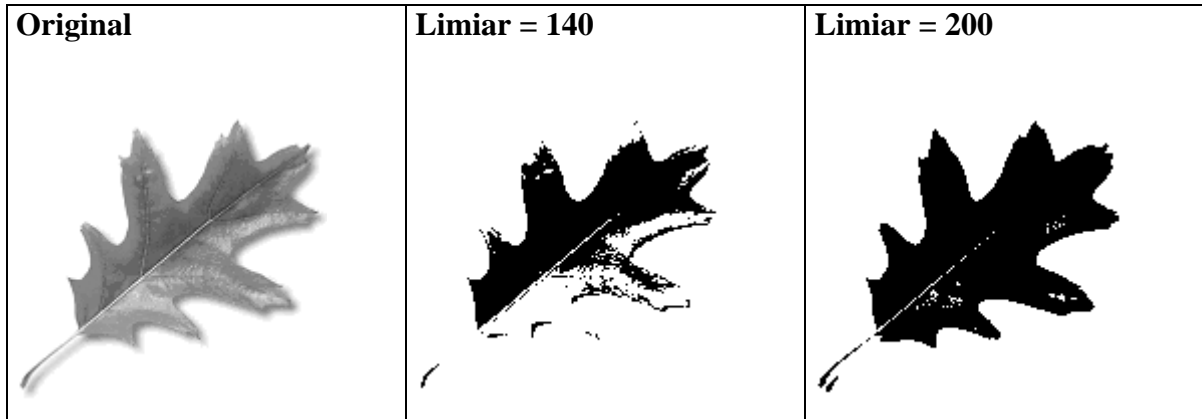


Figura 2.2. Exemplo de imagem com variação do Limiar.

2.3.2 Realce

O realce das imagens tem por finalidade destacar determinadas características da imagem, tendo por objetivo evidenciar detalhes da imagem para uma aplicação específica (Gonzalez *et al.*, 2003). As técnicas de realce podem se basear em processamento ponto-a-ponto, onde o novo valor dos pixels analisado independe dos pixels vizinhos, ou pelo processo de filtragem, onde o pixel analisado irá ter seu valor definido levando em consideração o valor dos pixels vizinho.

Uma das técnicas utilizadas na computação gráfica para realçar o contraste de objetos da imagem é o histograma, que contabiliza a quantidade de *pixels* para cada nível de cinza de um espectro (Rodrigues, 2003). Segundo (Felau, 2008), para melhorar o contraste da imagem devemos realçar as partes mais escuras, tornando-as mais claras, e escurecendo as partes mais claras.

A equalização de uma imagem consiste na mudança do valor de cada *pixel* visando uniformizar a distribuição de níveis de cinza, melhorando a iluminação da imagem (Santos, 2006).

2.3.3 Definição do Histograma de uma Imagem

O histograma é utilizado para verificar a distribuição dos níveis de cinza de uma imagem, o mesmo fornece a informação de quantos pixels na imagem possuem um determinado nível de cinza, pode verificar se a imagem está com alto ou baixo contraste (Filho, 1999). Em geral o histograma é visualizado através de um gráfico (x,y), onde dizemos que x é a distribuição dos níveis de cinza e y é a frequência que os mesmos ocorrem (figura 2.3 e 2.4).

De acordo com (Gonzaga, 1993), o histograma não traz informação posicional sobre os pixels da imagem. Mas é uma ferramenta indispensável na etapa de pré - processamento, pois fornece a visão estática dos pixels, sobre o contraste da imagem e os níveis de iluminação. Imagens defeituosas, por exemplo, uma fotografia com iluminação mais fraca ou mais forte, pode ser melhorada através de uma modificação em seu histograma.

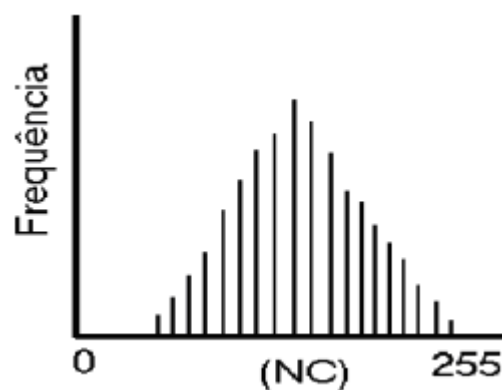


Figura 2.3. Exemplo de histograma.

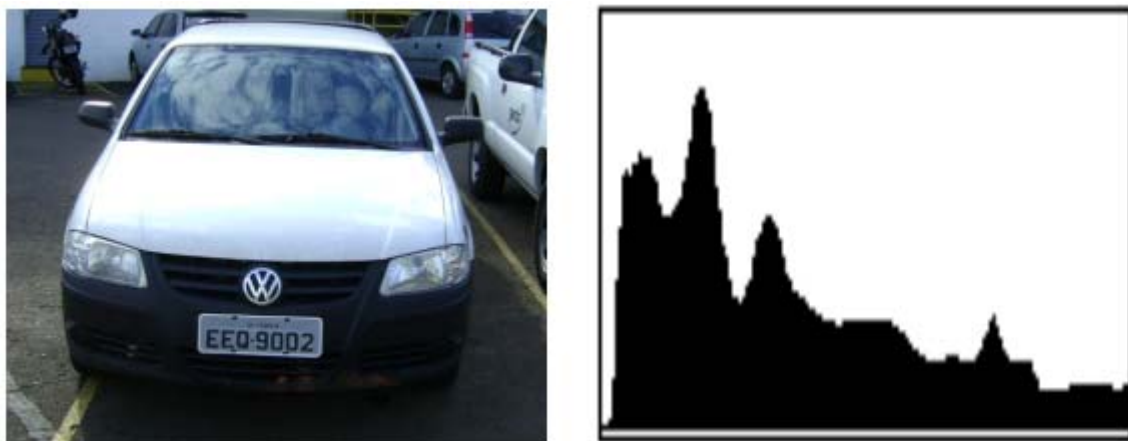


Figura 2.4. Imagem digital e seu histograma.

O histograma é bastante utilizado na etapa de segmentação, principalmente em técnicas que utilizam similaridade entre os pixels (Souza, 2000).

Se o histograma obtiver uma concentração de pontos á esquerda do gráfico a imagem é escura (figura 2.5), por outro lado, se no histograma verificarmos os pontos bem distribuídos a imagem possui um bom contraste (figura 2.6).



Figura 2.5. Imagem escura e seu histograma.

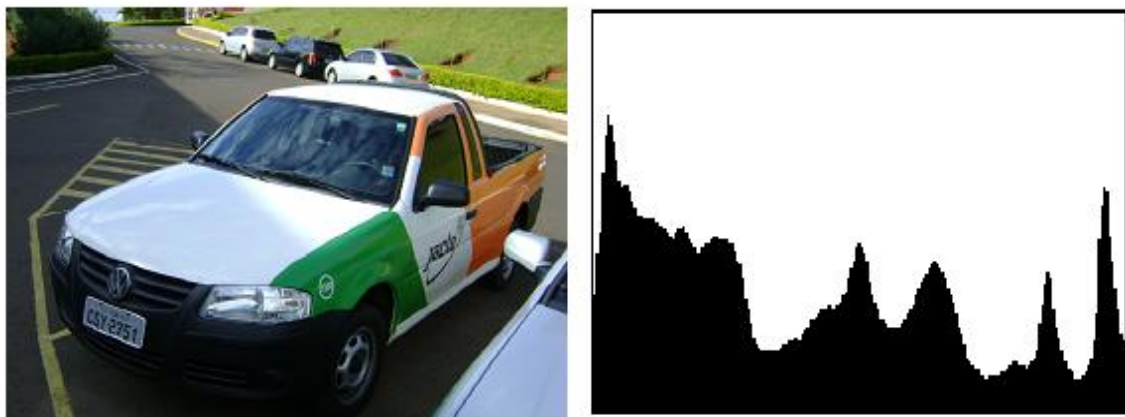


Figura 2.6. Imagem com bom contraste e seu histograma.

2.3.4 Bordas

De acordo com (Gonzalez *et al.*, 2003), uma borda é o limite entre duas regiões com propriedades relativamente distintas de nível de cinza. (Neves *et al.*, 2001), afirmam que para detecção de bordas têm-se dois tipos de filtros, os gradientes baseado na luminosidade da imagem e o Laplaciano que é uma função derivada de segunda ordem.

A detecção de bordas é muito importante no processo de análise de imagem devido a necessidade de se identificar e classificar os objetos que formam uma imagem.

Existem várias possibilidades de detecção de borda que utilizam o conceito de gradiente, citamos: Canny, Sobel, Roberts e Prewitt.

A figura abaixo ilustra a aplicação do filtro de detecção de bordas Sobel em uma imagem.



Figura 2.7. (a) Imagem Original, (b) Imagem aplicado o filtro Sobel

2.3.5 Limiarização

A limiarização é uma técnica de processamento de imagem que consiste em agrupar pixels da imagem com tons que sejam semelhantes, utilizadas para separação dos objetos com o fundo da imagem. A aplicação depende da definição de um limiar T que serve como divisor entre os pixels, os valores superiores ao limiar recebem valor máximo e os inferiores recebem valores mínimos, fazendo a binarização da imagem.

A operação de limiarização simples pode ser descrita matematicamente como uma técnica na qual uma imagem de entrada $f(x, y)$ de N níveis de cinza produz à saída uma imagem $g(x, y)$, chamada de imagem limiarizada, cujo número de níveis de cinza é menor que N . A imagem $g(x, y)$ apresenta dois níveis de cinza, sendo

$$g(x, y) = \begin{cases} 0, & \text{se } f(x, y) < T \\ 1, & \text{se } f(x, y) \geq T \end{cases}$$

Fórmula 2.2. Limiarização simples

onde T é um valor de tom de cinza predefinido denominado limiar, os pixels rotulados com 0 correspondem ao fundo e os rotulados com 1 correspondem aos objetos.

A limiarização pode ser dividida em global ou local. A limiarização global é feita utilizando um único limiar para toda a imagem, enquanto, que a limiarização local utiliza para cada área distinta da imagem um valor de limiar diferente.

O resultado da limiarização de uma imagem está relacionado com a definição correta de um limiar. Diversos são os métodos utilizados para tentar encontrar um bom valor para o limiar, os mais utilizados são baseados na equalização do histograma da imagem e em métodos estatísticos como a média e mediana da vizinhança diagonal dos *pixels* que formam a imagem (Gonzalez *et al.*, 2003).

Na figura 2.8 pode-se ver a imagem com um fundo contrastante e seu histograma.

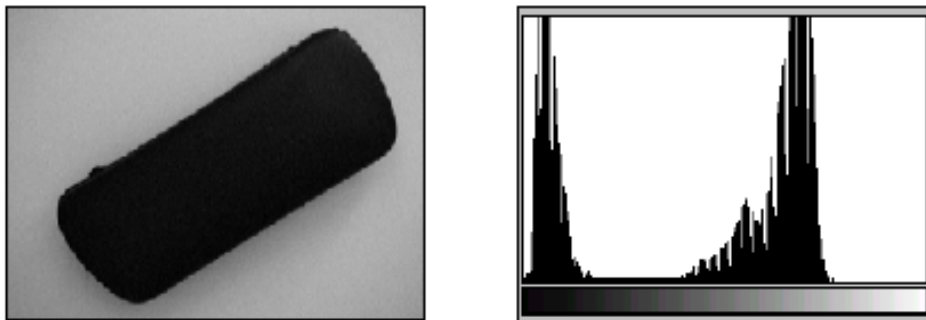


Figura 2.8. Imagem e seu histograma (Souza, 2000).

Na figura 2.9 foi escolhido um limiar global e foi realizado a limiarização global da imagem, pode-se ver que a segmentação da imagem foi bem sucedida.

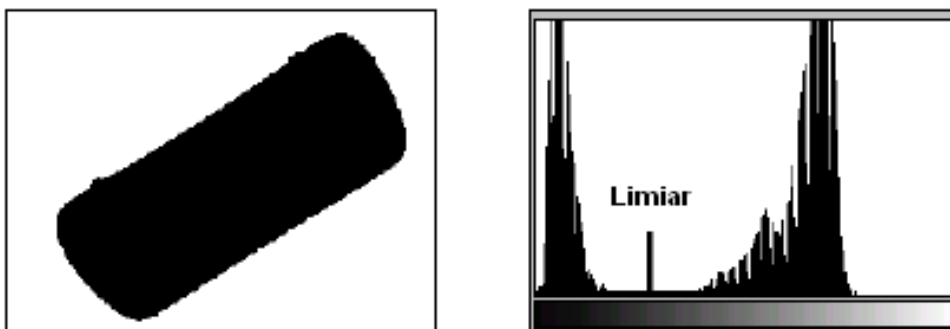


Figura 2.9. Imagem segmentada e escolha do limiar global (Souza, 2000).

Na limiarização local a imagem é dividida em partes que possuem propriedades comuns, como brilho e cor, e um limiar local é escolhido para cada parte. Na figura 2.10 pode-se perceber que a imagem possui iluminação diferente em determinadas partes, então a escolha de um limiar global não segmentará a imagem com o resultado pretendido.

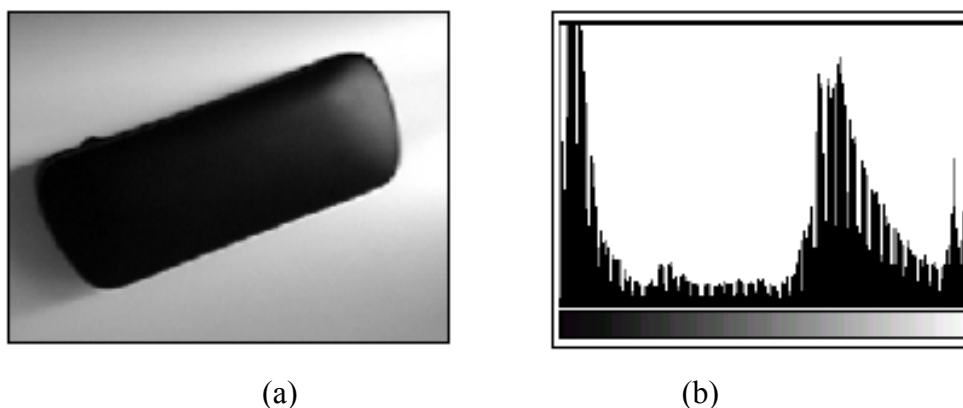


Figura 2.10. (a) Imagem, (b) Histograma (Souza, 2000).

A Figura 2.11 mostra a segmentação da imagem e o valor do limiar global escolhido, pode-se perceber que a segmentação realizada não obteve o resultado pretendido devido à utilização da limiarização global ou a escolha do limiar.

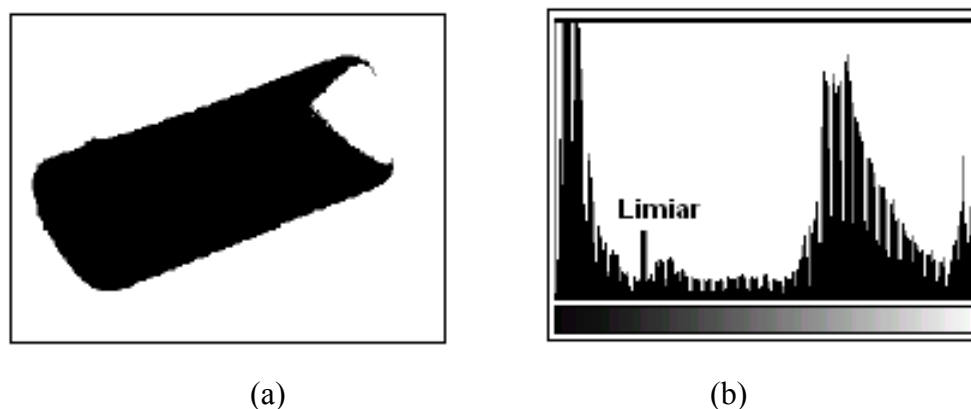


Figura 2.11. Imagem segmentada e escolha do limiar global (Souza, 2000).

2.4 Elementos de um Sistema de Processamento Digital de Imagens

O Sistema de processamento de imagens é descrito por 5 elementos (González *et al.*, 1993).

(1) – Aquisição de Imagem: Os computadores somente podem processar imagens codificadas em informações digitais, e as imagens podem ser encontradas em diversas formas e o pré-requisito para o processamento digital é a conversão para o formato digital (Castleman, 1996).

Um digitalizador de imagens deve ter a capacidade de dividir uma imagem em pixels e endereçar individualmente para medir a quantidade de energia em cada pixel, quantificar a medição contínua para produzir um conjunto de valores inteiros, e escrever este conjunto em um dispositivo de armazenamento de dados. Para ser possível um digitalizador precisa ter cinco elementos (Castleman, 1996).

Os elementos são: sistema de amostragem, mecanismo de varredura da imagem, sensor de luz, um quantificador e um armazenador.

(2) – Armazenamento: O armazenamento é medido por bytes, kbytes, mbytes, gbytes e tbytes.

(3) – Processamento: Procedimento que deve ser em forma de algoritmo, com exceção da aquisição e apresentação da imagem, a maioria das funções de processamento de imagem pode ser implementado em software (Gonzales *et al.*, 1993).

(4) – Comunicação: Envolve comunicação local, entre o sistema de processamento de imagem e comunicação remota de um ponto ao outro, tipicamente em conexões com transmissão de dados das imagens.

(5) - Apresentação: Item final em todo o sistema de processo de imagem. Os elementos básicos de um sistema de processamento de imagens citados por (Gonzalez *et al.*, 1993) são:

- Comunicação: Aquisição de imagem (vídeo e scanner) e Dispositivos de exibição (monitores, impressoras e máquinas fotográficas).
- Processamento: Para efetuar um processamento usar um computador.
- Armazenamento: Discos ópticos, fitas magnéticas, videotape, discos magnéticos.

2.5 Análise de Imagem

A análise é a etapa responsável por separar da imagem os elementos interessantes para a aplicação, especificamente o caractere. A análise da imagem é realizada em duas etapas, a segmentação e descrição (Souza, 2000).

Para separar as partes que são interessantes em uma imagem realizamos a segmentação, sendo esta responsável por dividir a imagem e mostrar para o sistema. Esse processo pode ser repetido muitas vezes, cada vez retirando da imagem elementos que não interessam, sendo o objetivo principal, isolar o elemento procurado.

De acordo com (Souza, 2000) o algoritmo de segmentação, para as imagens monocromáticas, é baseado em descontinuidades ou similaridades entre pixels. O algoritmo faz a segmentação da imagem verificando a mudança de níveis de cinza, sendo muito utilizada para a detecção de pontos isolados, linhas e bordas da imagem. Em seguida, faz a segmentação verificando regiões da imagem que possuem o mesmo padrão, como cor ou textura. De acordo com (Gonzalez *et al.*, 1993) esses algoritmos devem utilizar-se técnicas de limiarização, crescimento, divisão e fusão de regiões.

Após a realização da segmentação a imagem deve ser representada para que elemento encontrado para que possa ser analisado posteriormente. A representação pode ser feita através da representação das características externas (como bordas), ou de suas características internas.

A característica externa é geralmente utilizada quando o formato do elemento é necessário, e a interna quando o interesse está em sua textura, cor e outras características.

3 Sistemas de Reconhecimento

3.1 Introdução

Segundo (Dias, 2007), sistemas de reconhecimento de placas de carro são aqueles que automaticamente identificam um veículo a partir de uma imagem contendo a sua a placa. Atualmente estes sistemas são baseados em visão computacional, onde o veículo é identificado pela própria placa, por meio de processamento de imagem e técnicas conhecimento de caracteres alfanuméricos.

O processo de reconhecimento da placa de um carro tem início com a obtenção de uma imagem digital contendo o carro e sua respectiva placa, podendo esta imagem ser captura por câmera digital. Em seguida, localiza na imagem a região que contém a placa, separa os caracteres que formam a placa e os reconhece. Para realização de cada processo são utilizados cálculos matemáticos, inteligência artificial e técnicas de segmentação imagem. Como saída do processo de reconhecimento, são disponibilizado os caracteres que compõem a placa no formato ASCII, possibilitando a utilização dos mesmos em qualquer outro processo computacional (criação de um registro de entrada, uma busca em um banco de dados de veículos, etc.). A figura 3.1 ilustra o diagrama de blocos básico de um sistema de reconhecimento de placas de automóveis

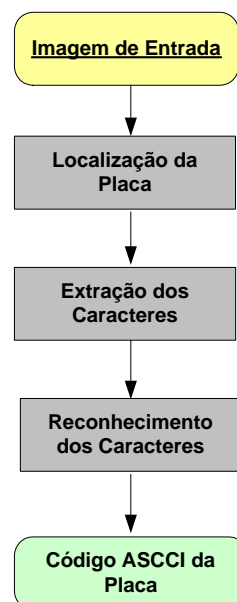


Figura 3.1. Estrutura do sistema de reconhecimento da placa (Dessbesell, 2008)

Como o objetivo do trabalho é estudar algumas teorias existentes necessárias para construção do sistema de automatização para o controle de veículos na portaria de entrada da empresa Jacto S.A. Este capítulo apresenta alguns procedimentos e técnicas que podem ser utilizadas para atingir este objetivo.

3.2 Localizações da Placa

Nesta etapa, é realizada uma busca pela região da imagem digital onde a placa do carro se encontra (figura 3.2). A busca é feita levando em consideração características inerentes da placa do carro, como contraste, textura e geometria. (Dessbesell, 2008) afirma que a etapa de localização é um ponto crítico do processo de reconhecimento de placas de carros, pois a não localização de pelo menos uma região contendo a placa implica na falha do processo como um todo.



Figura 3.2. Região candidata encontrada durante a localização

Dentre os métodos que existentes empregados para esta tarefa são citados por (Dias, 2005) o Método de Morfologia dos caracteres, o qual se baseia na geometria dos mesmos; o Método de variação Tonal, que vasculha as linhas em busca de um padrão de frequência e amplitude (assinatura); o Método de Correlação, que busca correlacionar a sub-imagem do objeto procurado na imagem pesquisada; O método da Detecção das Arestas, que emprega detectores de contorno para localizar as quatro arestas da placa; o Método Conexionista, que utiliza técnicas conexionistas na busca por regiões cuja cor e

textura sejam similares ao de uma placa do carro; o Método Híbrido, que emprega uma combinação de dois ou mais métodos distintos, visando aumentar a eficiência do processo.

Dentre os inúmeros trabalhos encontrados na literatura, a seguir, destaca-se aqueles que possuem alguma influência sobre o sistema desenvolvido nesse trabalho.

No ano de 1997, Barroso *et al.* propuseram um algoritmo para encontrar a “assinatura” padrão. O algoritmo procura um conjunto de máximos e mínimos contínuos que possuem algumas características predefinidas (amplitude, número, distância, etc.)

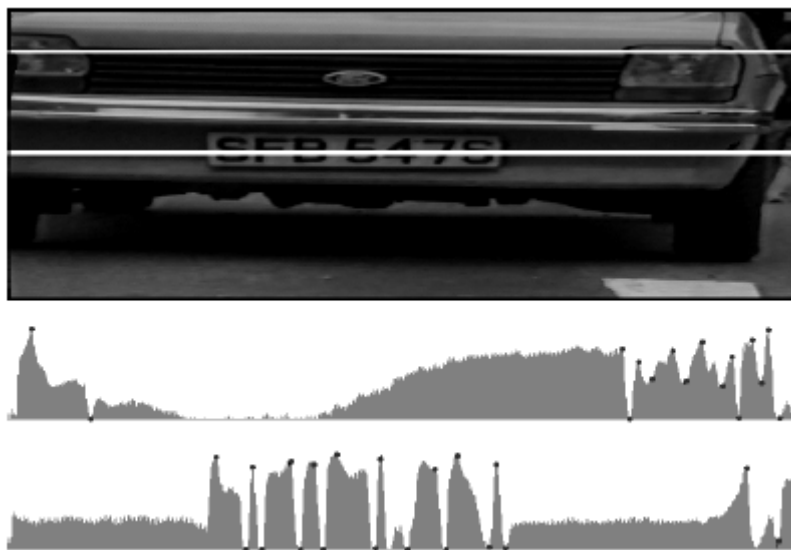


Figura 3.3. Corte transversal da imagem do carro (Barroso et al, 1997)

(Souza, 2000) realiza a localização da placa através de um algoritmo de análise por variação tonal, onde é realizada uma busca em toda a imagem a procura de linhas horizontais que possuam um padrão tonal diferente do restante da mesma (figura 3.3 (a) e (b)). Este padrão de variação fornece uma “assinatura” que permite um bom desempenho na localização da placa. Para verificar se a linha encontrada corresponde à placa (Souza, 2000) restringe a imagem a uma janela de 120 x 50 pixels, onde possivelmente estará a placa e onde o processo de verificação será executado, aumentando o desempenho dos algoritmos utilizados. Após esta análise, a região encontrada é binarizada para ser analisada por algoritmo de procura e verificação de dígitos.

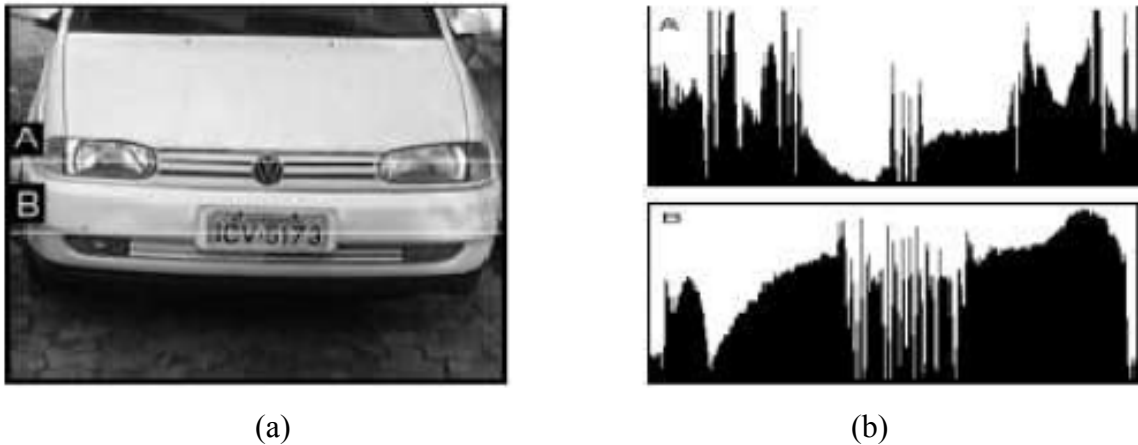


Figura 3.4. (a) Imagem original, (b) Variação tonal das linhas A e B (Souza, 2000).

O processo de verificação é realizado em três partes: binarização, busca por dígitos e verificação dos grupos válidos, que são descritas a seguir.

3.3 Processos de Verificação da Placa e Extração de Caracteres

Depois de localizada a região da imagem que supostamente possui placa, a mesma é vasculhada em busca dos caracteres que formam esta placa. Dentre os métodos existentes empregados para localização desta tarefa são citados por (Dessbesell, 2008) o Método das Projeções que define os limites de cada caractere por meio de projeções horizontais e verticais; o Método da Morfologia, o qual separa os caracteres com base em seus componentes conexos; o Método da Caixa Limitante Adaptativa, que emprega uma construção no formato de “L invertido” para, num primeiro momento encontrar os limites esquerdo e superior dos caracteres e após, os demais limites; e o Método de Etiquetagem dos componentes Conectados que atribui uma etiqueta única aos pixels que compõem um mesmo objeto.

Segundo (Souza, 2000), buscando eliminar as diferentes variações de iluminação e facilitando o trabalho do algoritmo de localização dos caracteres, a primeira técnica utilizada deverá ser utilizada é o algoritmo de binarização, tendo como resultado uma imagem binária.

Entre as técnicas para localizar os caracteres existentes na literatura, destacamos aqueles que possuem alguma influência sobre o sistema desenvolvido nesse trabalho.

(Belvisi et al, 1999), utilizando a suposta posição da placa encontrada pela variação tonal, apresentou um algoritmo que realiza uma busca na imagem a procura de uma área quadrada fechada, no tamanho aproximado da placa, conforme é mostrado na figura 3.5. Em seguida é realizada a projeção horizontal e vertical das letras (figura 3.6), que consiste na contagem dos pixels existentes em uma linha ou coluna respectivamente.

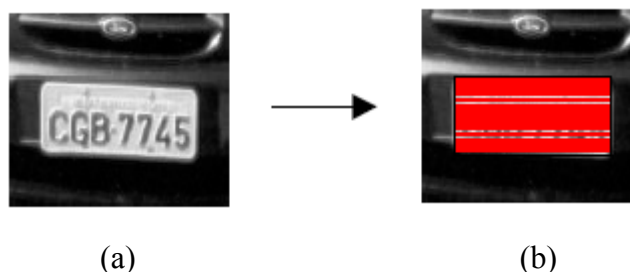


Figura 3.5. (a) Imagem original; (b) Imagem da placa encontrada (Belvisi *et al.*, 1999).

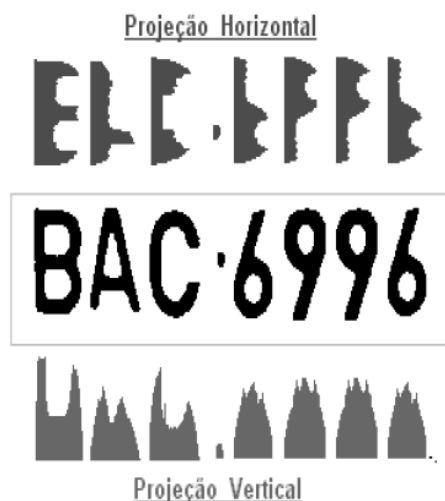


Figura 3.6. Projeção horizontal e vertical (Belvisi *et al.*, 1999)

Após a realização das projeções, a segmentação dos caracteres é feita através de uma varredura vertical de forma a localizar o início e o fim de cada caractere, como mostra a figura 3.7.

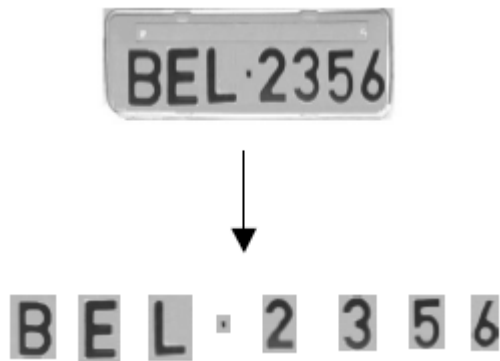


Figura 3.7. Segmentação dos caracteres (Belvisi *et al.*, 1999).

(Souza, 2000) utiliza para localizar os caracteres a técnica de procura por padrões que a partir da posição da placa encontrada na variação tonal realiza um algoritmo dividido em três estágios.

- A imagem é analisada a procura de uma região com o formato L invertido, cujos pixels nela contidos sejam todos brancos (estágio 1).
- Uma barra horizontal é deslocada de cima para baixo à procura do limite inferior do caractere (estágio 2).
- Uma barra vertical é deslocada da esquerda para direita em busca do término do caractere (estágio 3).

Após a execução dos três estágios, são selecionados elementos encontrados que possuem as dimensões horizontais e verticais correspondentes as de um caractere (figura 3.8). Em seguida, é verificado se as regiões encontradas formam um grupo válido para ser considerado como pertencente à placa de um veículo. Os grupos devem conter características existentes em uma placa, como sete dígitos alinhados, para isso é analisado a disposição espacial e a dimensão desses elementos. Caso não seja localizado um grupo válido, é realizada novamente a binarização da imagem, mas com outro fator de referência, e então realizados os processamentos subseqüentes.

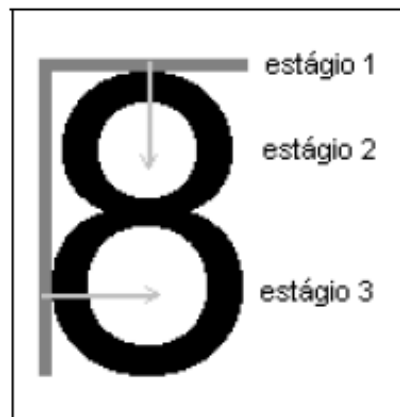


Figura 3.8. Estágios utilizados na busca por padrões (Souza, 2000).

3.4 Técnicas de Reconhecimento de Caracteres

Dentre os métodos que podem ser empregados no reconhecimento de caracteres são citados por (Dessbesell, 2008) o Método Estatístico, que emprega uma medida de distância na procura da amostra de maior similaridade em um conjunto de amostras rotuladas; o Método Conexionista, o qual utiliza um conjunto de amostras rotuladas para treinar uma rede e elementos, tendo fronteiras de decisão não-lineares definidas iterativamente pelo método de treinamento e pelo conjunto de amostras; e o Método Sintático, que se baseia na estrutura dos padrões de entrada, onde uma gramática é empregada na definição das regras de formação de tais estruturas e um autômato utilizado na operacionalização da tarefa.

3.4.1 Redes Neurais Artificiais

Redes neurais são funções matemáticas capazes de aprender o comportamento de funções do ambiente. Estas redes foram propostas inicialmente por (McCulloch *et al.*, 1943), a partir do estudo do comportamento dos neurônios do cérebro. Um modelo geral de um neurônio artificial é apresentado na figura 3.9.

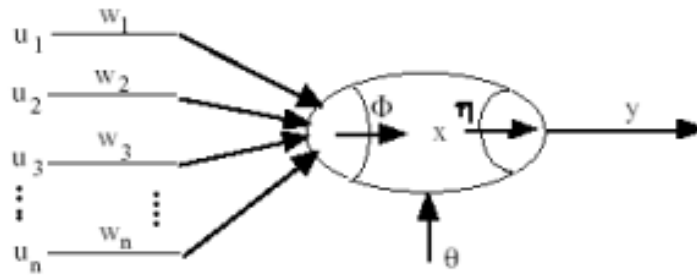


Figura 3.9. Modelo geral de um neurônio artificial proposto por (McCulloch *et al.*, 1943).

Para produzir um estado de ativação do neurônio as entradas u_i e w_i serão combinadas usando a função ϕ . A função η vai produzir a saída do neurônio, onde um valor abaixo θ a saída será nula.

Em geral Redes Neurais Artificiais são caracterizadas por: um grande número de elementos simples de processamento (nós); um grande número de elementos de conexão entre os nós (pesos), os quais armazenam o conhecimento adquirido; alto paralelismo, controle distribuído e habilidade para aprender automaticamente as representações internas (atualizações das interconexões). Os nós de processamento executam a soma dos pesos de suas entradas, efetuando e obtendo um simples valor de saída através de alguma função de ativação não linear. Os nós são ativados se o valor da função de ativação for maior que um determinado valor de limiar ajustável, associado a cada nó (Minsky *et al.*, 1969).

A topologia da rede é definida como a forma em que a rede está organizada, sendo o conhecimento armazenado nos pesos das interconexões. Redes neurais são classificadas segundo sua topologia, dividindo-se em: redes supervisionadas e não supervisionadas. Redes supervisionadas são treinadas através da minimização do erro entre a saída desejada e a saída obtida, e quando este erro é menor que um limiar, a rede está devidamente treinada (Kovács, 1996). Ainda no círculo das redes supervisionadas, existem as redes recorrentes, que necessitam dos dados previamente estabelecidos, mas não necessitam de uma saída exata.

As redes não supervisionadas criam agrupamento de dados, ajustando automaticamente as representações dos pesos internos a partir de uma determinada entrada, não havendo a necessidade de um supervisor. Exemplos de redes neurais não supervisionadas podem ser encontrados em (Barreto *et al.*, 1999).

Redes não supervisionadas são interessantes por aprenderem o comportamento de um ambiente multivariável baseados na sua dinâmica. Uma das redes neurais amplamente utilizadas nesta forma de aprendizagem, sendo usada também como geradora de previsões são os modelos de redes neurais auto-organizáveis, onde os pesos são ajustados à medida que os agrupamentos de dados vão sendo formados. Modelos de simulação de séries temporais e previsão usando redes auto-organizáveis podem ser encontradas nos trabalhos de (Barreto *et al.*, 2000) e (Barreto *et al.*, 2001).

Entre os textos existentes na literatura que utilizam redes neurais na identificação de caracteres alfanuméricos, apresentamos, a seguir, um breve comentário do trabalho realizado no ano de 2002 por Vásquez *et al.*.

(Vásquez *et al.*, 2002) propõe um sistema de identificação de placas de carro que utiliza a técnica de redes neurais multicamadas para o reconhecimento dos caracteres que formam a placa. O sistema proposto consiste de dois processos: o processo de treinamento e do processo de reconhecimento (figura 3.10).

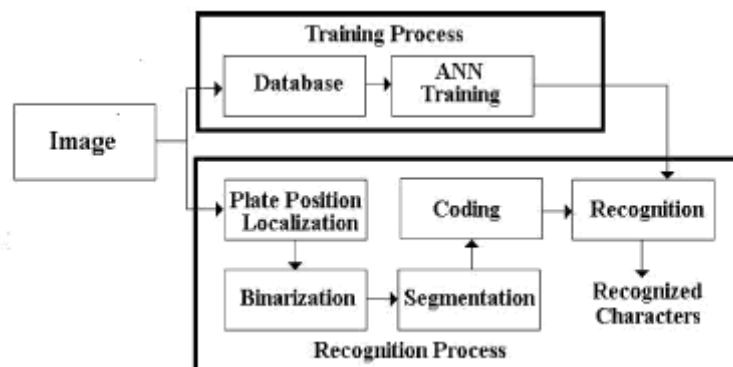


Figura 3.10. Sistema proposto para reconhecimento de veículos (Vásquez *et al.*, 2002).

O processo de treinamento consiste, principalmente, do desenvolvimento de banco de dados das placas de caracteres codificados e da criação de uma rede neural multicamadas utilizadas na identificação. A codificação encontrada é inserida em uma base de dados que contém o caractere e sua codificação, sendo usada posteriormente no processo de reconhecimento. Para que a fase de codificação seja realizada deve-se passar pelas etapas de: Captura de imagem; Segmentação dos símbolos; Binarização; Redução de ruídos; por fim Codificação. As fases de captura de imagem, segmentação dos símbolos, binarização e redução de ruídos já foram comentadas nas seções anteriores.

Para se achar a codificação do caractere doze linhas horizontais são traçadas como mostra na figura 3.11. Cada uma delas composta de seqüências de pixels brancos (valor 1) e seqüências de pixels pretos (valor 0), o que gera um vetor de valores binários, sendo que o tamanho do vetor será o número de pixels que a linha conterá. Assim, para cada padrão, 12 vetores são obtidos.

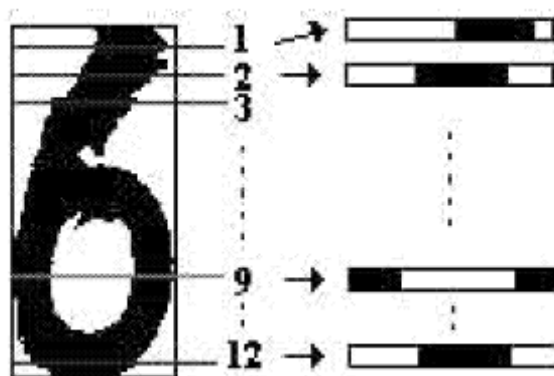


Figura 3.11. Doze códigos de vetores obtidos por doze linhas horizontais (Vásquez *et al.*, 2002)

Antes de comparar as seqüências com a tabela de binários usada para atribuir um código a cada seqüência, um processo de redução de ruído deve ser feito, pois, o ruído introduzido pela aquisição e processos binarização pode introduzir erros durante o processo de codificação. Para isso deve-se analisar a quantidade mínima de zeros ou uns juntos que o vetor poderá ter. A figura 3.12 mostra um exemplo onde existe uma detecção de ruídos, o símbolo em vermelho foi detectado com ruídos e foi invertido o seu valor.

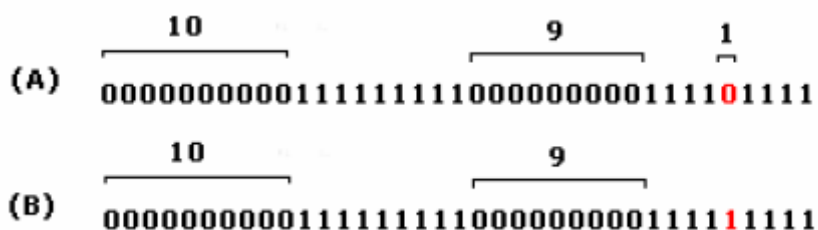


Figura 3.12. Exemplo de detecção de ruídos (Vásquez *et al.*, 2002)

Depois que o ruído é reduzido cada vetor gerado no processo anterior é comparado com os vinte e três códigos da tabela de códigos (figura 3.13), suas linhas

são codificadas correspondentes às combinações das seqüências em preto-branco. Para montar a tabela de códigos foram utilizados dois símbolos: o ‘**n**’, que significa a cor preta, e o ‘**b**’, que significa a cor branca. A combinação de ‘**n**’ e ‘**b**’ forma uma tabela de códigos, como a da figura 3.13. O símbolo ‘ \bar{n} ’ significa que a seqüência da cor preta tem um comprimento mais longo e o símbolo \bar{b} significa que a seqüência da cor preta tem um comprimento mais longo. Por exemplo, ‘**n + b + n**’ significa que a combinação consiste de uma seqüência de preto, branco e preto. Assim, se olharmos para a tabela de códigos, os códigos 8 e 9 têm a mesma seqüência, mas no código de 8 a primeira seqüência em branco é mais longa, enquanto no código 9 é a segunda em branco que é mais longa. Assim, de acordo com (Vásquez *et al.*, 2002), essa codificação permitir distinguir entre os caracteres com algumas semelhanças, tais como, 1 e 7 ou U e V.

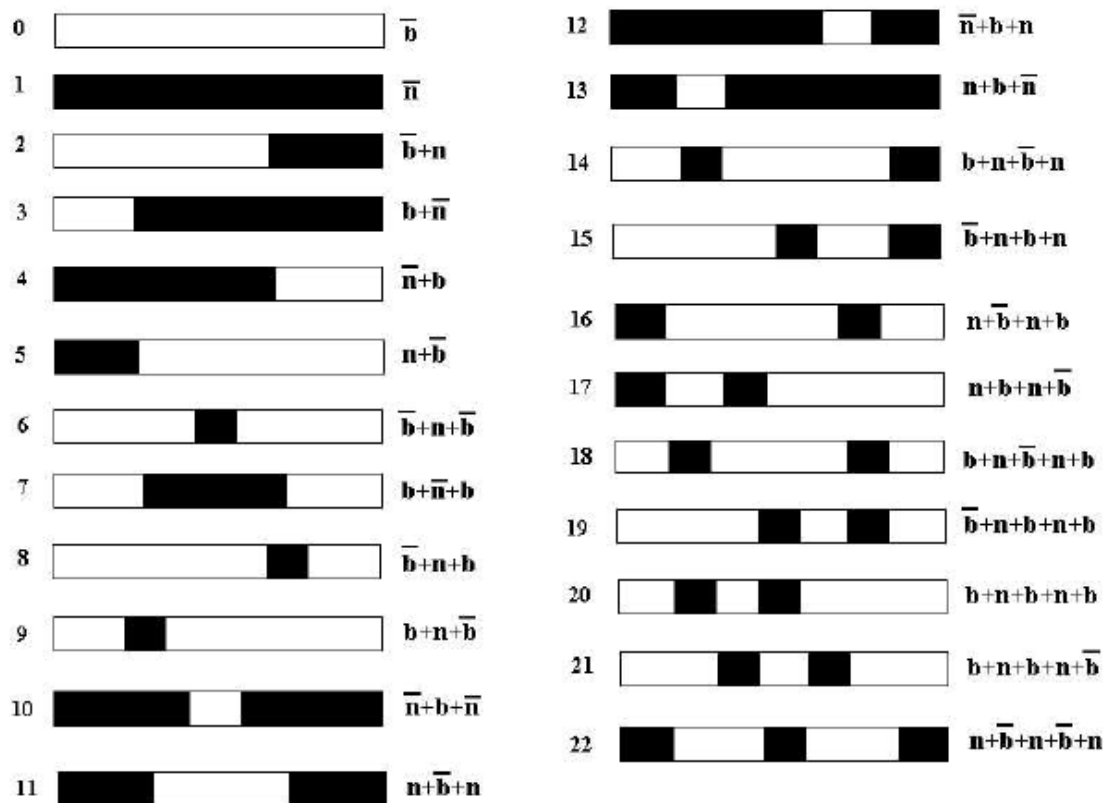


Figura 3.13. Tabela de códigos utilizada para codificação de caracteres da placa
(Vásquez *et al.*, 2002).

(Vásquez *et al.*, 2002) afirmam que as diferenças entre as imagens do mesmo caractere, obtidas de placas diferentes, devem ser usadas para treinar o algoritmo e fazer com que ele seja o mais preciso possível. A figura 3.14 mostra um exemplo de quatro imagens diferentes do mesmo caractere que são utilizadas para treinar a rede proposta em (Vásquez *et al.*, 2002)



Figura 3.14. Caractere 7 obtido de quatro placas diferentes (Vásquez *et al.*, 2002).

A tabela 3.1 apresenta as seqüências de códigos obtidas após a aplicação do processo de codificação para os quatro caracteres mostrada na figura 3.14. Cada número corresponde a uma seqüência de códigos da tabela de códigos apresentada na figura 3.13. Para identificação do caractere é realizada uma comparação dos códigos obtidos com os existentes na base de dados gerada pelo treinamento. A comparação é feita de modo a obter um padrão que seja idêntico ou mais próximo do obtido após a segmentação

Tabela 3.1. Codificação da figura 3.14 (Vásquez *et al.* 2002)

Padrão	Códigos											
1	1	2	8	8	8	8	6	9	9	9	7	9
2	1	1	3	8	8	7	9	9	9	9	9	9
3	1	1	8	8	8	6	9	9	9	9	9	9
4	1	2	8	8	8	6	9	9	9	9	9	9

4 Implementação do Sistema Proposto

4.1 Introdução

No desenvolvimento do sistema proposto foi utilizada a técnica para a localização da placa com base na busca pela variação tonal correspondente a placa apresentado em (Barroso *et al.*,1997), o algoritmo de identificação de borda de Sobel (Souza, 2000), segmentação dos caracteres utilizando a técnica de limiarização, utilizada em (Souza, 2000) e o reconhecimento do caractere apresentado em (Vázquez *et al.*,2002).

Basicamente o software desenvolvido possui a função de abrir uma imagem que contém o carro e sua placa, que foi adquirida por uma câmera digital, localizar a região da imagem que contém a placa, segmentar os caracteres e identificá-los.

O sistema foi desenvolvido na plataforma Windows XP Professional, utilizando para testes o notebook HP DV600, com capacidade 2 GB de memória ddr2 e processador turion 64 2x 1.8 mhz. O ambiente de desenvolvimento é o Netbeans 6.0.1, banco de dados Firebird 1.5 e a tecnologia escolhida foi a linguagem de programação Java.

4.2 Java

Java é uma tecnologia desenvolvida pela Sun Microsystems que consiste uma linguagem de programação, uma plataforma que apóia a execução de sistemas desenvolvidos em Java e um conjunto de API que possui bibliotecas gráficas, para armazenamento de dados, processamento de dados, I/O e suporte de rede (Reilly *et al.*, 2002).

De acordo com (Reilly *et al.*, 2002) o início do desenvolvimento da linguagem Java ocorreu na década de 1990, onde uma equipe da Sun Microsystems começou a trabalhar para desenvolver uma linguagem de programação para o mercado de sistemas embutidos. As principais características de Java são a orientação a objetos, simplicidade, garbage collection, portabilidade, segurança.

O Netbeans é um ambiente de desenvolvimento integrado Java, multiplataforma, fornecendo uma base sólida para o desenvolvimento de projetos (Boudreau *et al.*, 2003).

4.3 Firebird

O Firebird é um gerenciador de Banco de Dados gratuito criado a partir do código fonte do Interbase. Sua vantagem em relação ao Interbase é a quantidade de recursos que os desenvolvedores acrescentaram em seu projeto (Leite, 2007).

4.4 Estrutura da Implementação.

Com base no trabalho (Manfio, 2008), o sistema desenvolvido utiliza o modelo de caso de uso representado pela Figura 4.1. Este modelo demonstra que o sistema poderá criar uma instância da classe e realizar duas funções, a aprendizagem ou o reconhecimento.

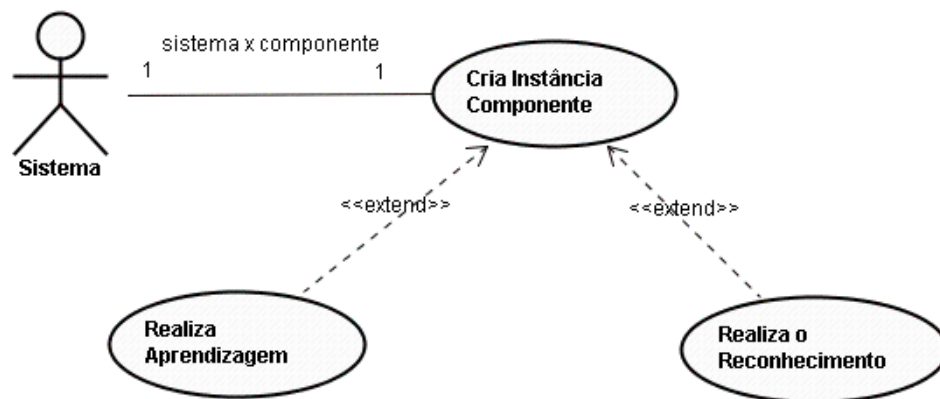


Figura 4.1. Modelo de caso de uso geral (Manfio, 2008).

Utiliza-se a função aprendizagem para capacitar a classe a reconhecer os caracteres da placa do veículo automotivo. A figura 4.2 é utilizada para demonstrar a necessidade de execução dos processos de receber a imagem, encontra a placa, segmentar os caracteres, mostra os caracteres e solicita ao usuário a identificação dos caracteres.

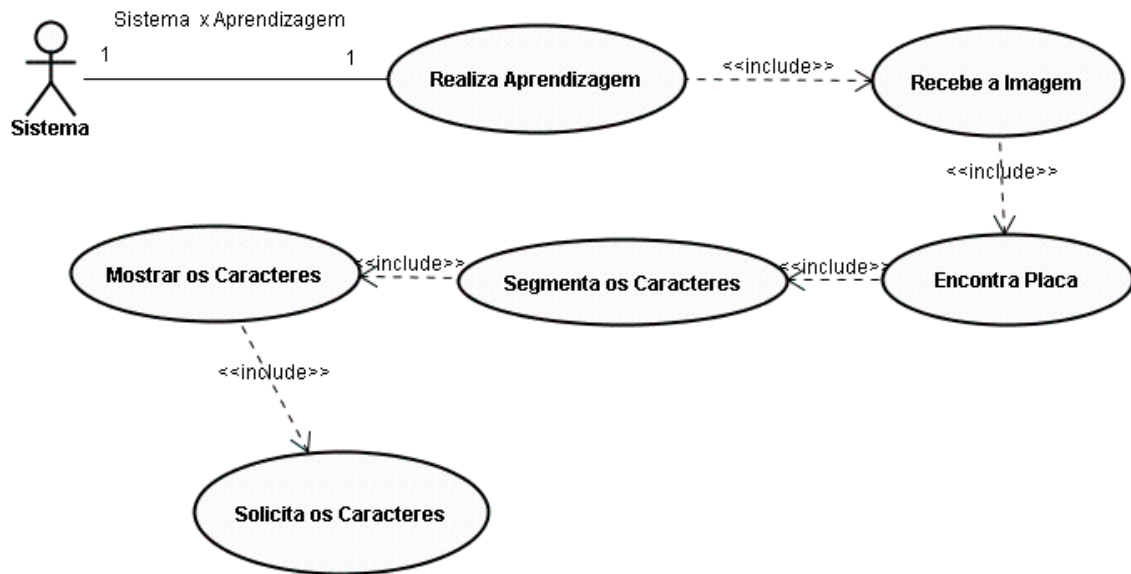


Figura 4.2. Modelo de caso de Aprendizagem (Manfio, 2008).

Tendo em vista o modelo de caso de uso representado na pela figura 4.3, que demonstra que é necessário realizar os processos de receber a imagem, encontrar a placa, segmentar os caracteres e identificar os caracteres. A função reconhecimento é utilizada para identificar automaticamente os caracteres da placa encontrada.

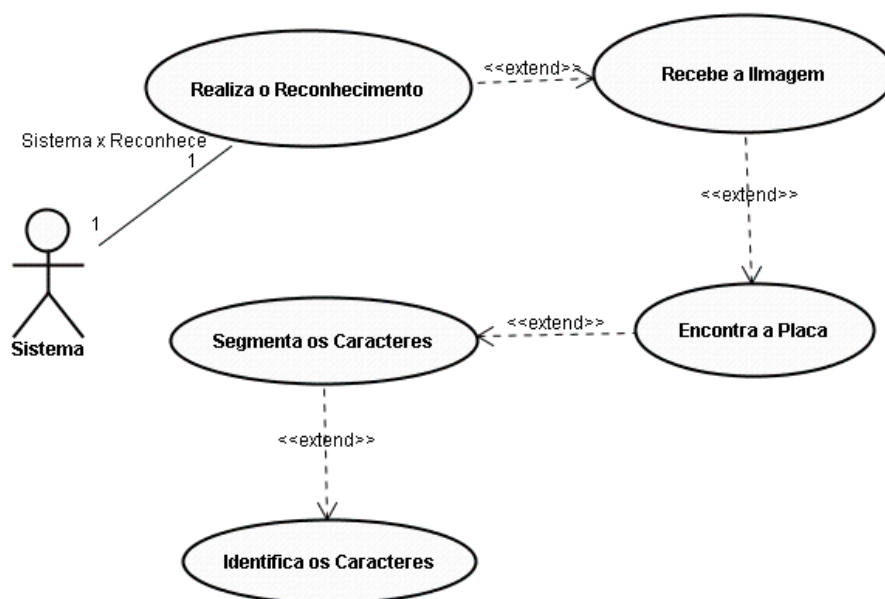


Figura 4.3. Modelo de caso de Reconhecimento (Manfio, 2008).

Com base no trabalho (Manfio, 2008) desenvolvemos o nosso sistema a partir do modelo de classes representado pela figura 4.4. Foram criados dois pacotes, um contendo os algoritmos de reconhecimento, chamado `j_reconhecer_placa`, e outro contendo a conexão com o banco de dados chamado de `bd_rede_neural`.

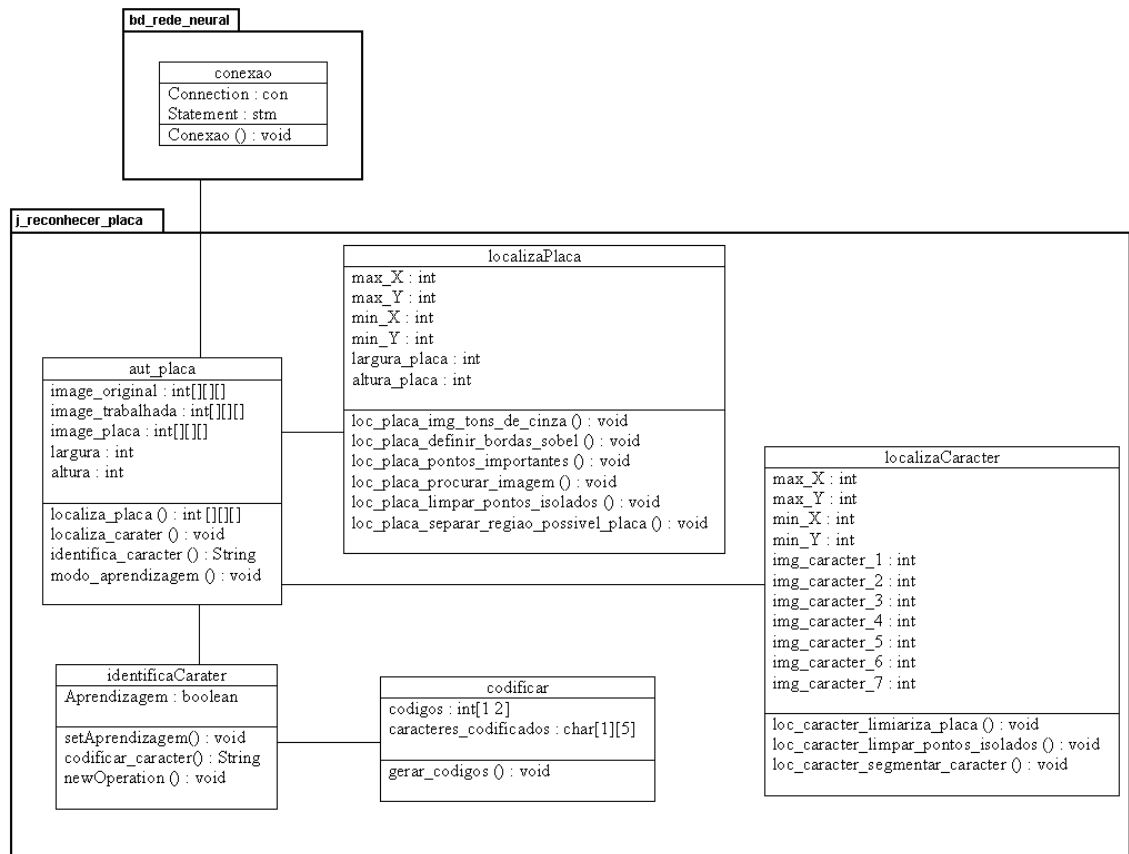


Figura 4.4. Modelo de Classe utilizado na implementação.

4.4.1 Pacote `j_reconhecer_placa`

O pacote chamado de `j_reconhecer_placa` contém as seguintes classes:

- `aut_Placa`.
- `localizaPlaca`.
- `localizaCaracter`.
- `identificaCaracter`.
- `codificar`.

Em todas as classes são implementados métodos para obter os resultados esperados.

O código fonte (figura 4.5) apresenta um parte do código fonte da classe `aut_Placa`. Estas classes contem todas as chamadas para encontrar a placa do carro e

reconhecer os seus caracteres, ainda a mesma é responsável pela comunicação com o usuário do software.

```

Placa_carro() {
    gerar_codigos();
}

public int[][][] localiza_placa() {
    loc_placa_img_tons_de_cinza();
    loc_placa_definir_bordas_sobel();
    loc_placa_pontos_importantes();
    loc_placa_procurar_imagem();
    loc_placa_limpar_pontos_isolados();
    loc_placa_separar_regiao_possivel_placa();
    return imagem_placa;
}

public void localiza_caracteres() {
    loc_caracter_limiariza_placa();
    loc_caracter_limpar_pontos_isolados();
    loc_caracter_segmentar_caracter();
}

public String identifica_caracteres() {
    setAprendizagem(false);
    return codificar_caracter();
}

public void modo_aprendizagem() {
    setAprendizagem(true);
    codificar_caracter();
}

```

Figura 4.5. Código fonte da classe aut_Placa.

O método `localizaPlaca` possui as seguintes funções:

- receber uma foto contendo o carro e sua placa;
- localizar a placa deste carro;
- segmentar a placa.

Este método foi baseado no trabalho de (Barroso *et al.*, 1997), o processo desenvolvido possui o seguinte conjunto de etapas: encontrar as bordas na imagem, definir, nas bordas encontradas, quais pontos são importantes, e procurar um conjunto de ponto que tem o formato e medidas de uma placa.

Baseado em (Souza, 2000), foi utilizado o algoritmo de Sobel (figura 4.6) para encontrar as bordas. Os parâmetros utilizados foram [0.1, 0.25, 0.75, -2.2, 0.75, 0.25, 0.1], sendo que caso o resultado da soma dos valores encontrados for menor que

zero o pixel não será considerado parte de uma borda e será indicado pela cor preta, mas caso o valor for maior que zero o pixel será indicado pela cor correspondente ao valor encontrado da soma (figura 4.7 - b).

```
private void enc_placa_definir_bordas_sobel(): () {
    int[][][] imageDataTemp = new int[largura][altura][3];
    double media;
    int cor;
    for (int h = 0; h < altura; h++) {
        media = 0;
        for (int w = 3; w < largura - 3; w++) {
            cor = imagem_trabalhada[w - 3][h][0];
            media = (cor * 0.1);
            cor = imagem_trabalhada[w - 2][h][0];
            media = media + (cor * 0.25);
            cor = imagem_trabalhada[w - 1][h][0];
            media = media + (cor * 0.75);
            cor = imagem_trabalhada[w][h][0];
            media = media + (cor * (-2.2));
            cor = imagem_trabalhada[w + 1][h][0];
            media = media + (cor * 0.75);
            cor = imagem_trabalhada[w + 2][h][0];
            media = media + (cor * 0.25);
            cor = imagem_trabalhada[w + 3][h][0];
            media = media + (cor * 0.1);
            media = Math.round(media);

            if (media < 0) {media = 0; }

            imageDataTemp[w][h][0] = (int) media;
            imageDataTemp[w][h][1] = (int) media;
            imageDataTemp[w][h][2] = (int) media;
        }
    }
    imagem_trabalhada = imageDataTemp;
}
```

Figura 4.6. Código fonte do algoritmo de Sobel

Logo após a aplicação do algoritmo de Sobel a imagem é binarizada (figura 4.7 – c), utilizando o limiar 50 conforme indicação do trabalho de (Souza, 2000), com o objetivo de obter os pontos importantes. Para as imagens que utilizamos nos testes do software os resultados utilizando limiar 50 foram satisfatórios.

Para localização da placa foi realizado uma procura dos conjuntos de pontos que caracterizam uma placa, conforme trabalho (Barroso *et al.*,1997). Toda imagem foi percorrida interligado os pontos importantes que possuem uma distância máxima entre eles (figura 4.7 - d). Por fim, é analisado o conjunto de pontos e identificado qual possui

tamanho proporcional ao de uma placa do carro, em seguida tal conjunto é segmentado na imagem (figura 4.7 - e).

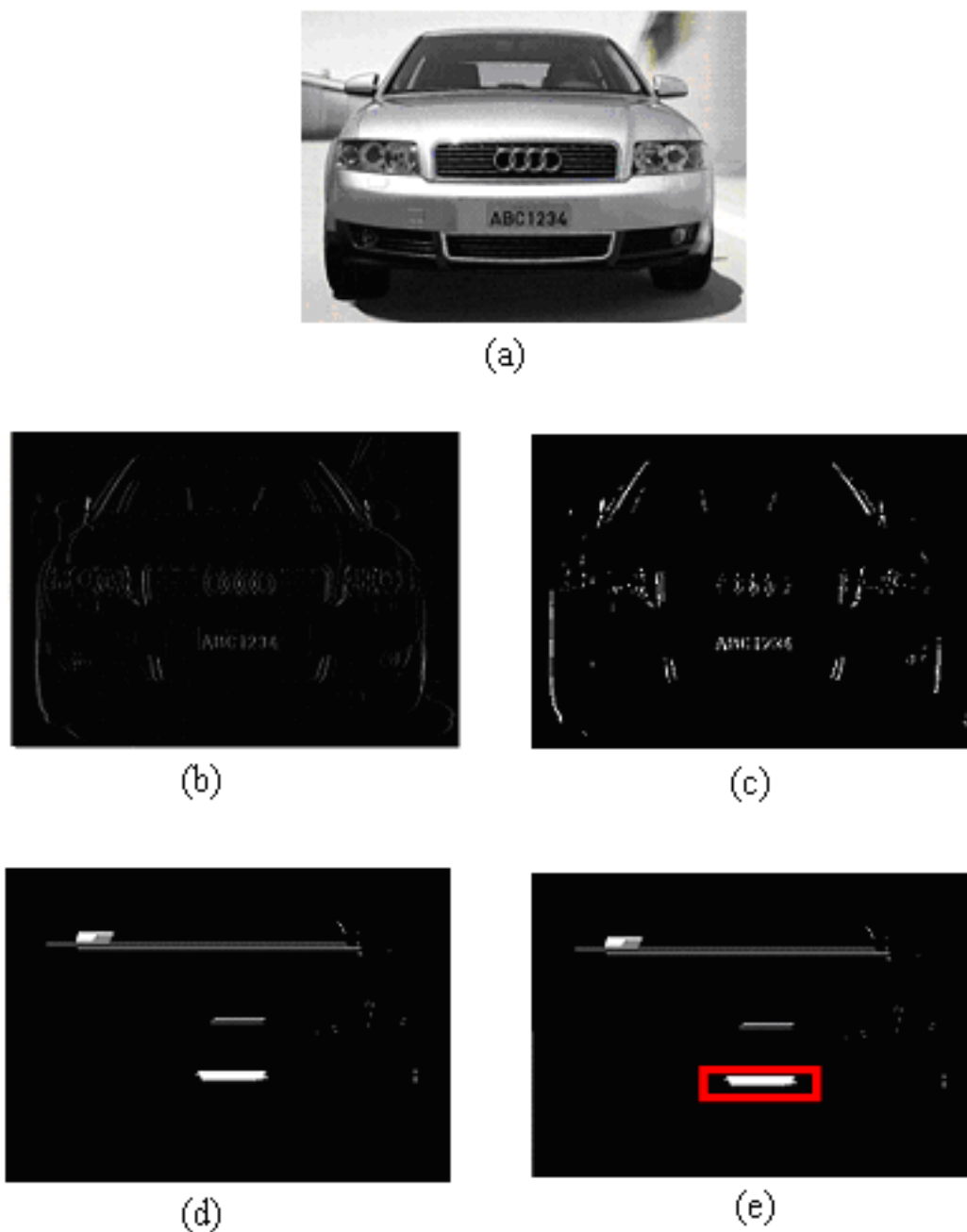


Figura 4.7. (a) Imagem original; (b) Imagem aplicado o algoritmo de Sobel (c) Imagem binarizada para definir pontos importantes; (d) Resultado da interligação do conjunto de pontos importantes; (e) Resultado da análise do conjunto de pontos e indicação da placa.

O método `localizaCaracter` é responsável por buscar os caracteres da placa após a localização da mesma pelo sistema. Este método utiliza a limiarização para eliminar as partes indesejáveis da placa, em seguida, seleciona os possíveis fragmentos da imagem que corresponda aos caracteres da imagem para iniciar a etapa de conhecimento (figura 4.8).



Figura 4.8. Caracteres encontrados pelo sistema

O método `modo_aprendizagem` e o método `identifica_caractere` foram desenvolvidos baseados na técnica de reconhecimento descrita por (Vàsquez et al., 2002) e nas sugestão de (Manfio, 2008). Um banco de dados foi construído contendo uma tabela com todas as combinações possíveis com os operadores ‘ ‘, ‘n’, ‘b’, ‘n̄’ e ‘b̄’.

Uma tabela, denominada “códigos_caracteres”, foi construída no Firebird 1.5 contendo todas as combinações possíveis com os operadores ‘ ‘, ‘n’, ‘b’, ‘n̄’ e ‘b̄’. A mesma é utilizada para armazenar as codificações dos caracteres encontrados nas placas dos veículos, sendo constituída das colunas: CARACTERE, COD1, COD2, COD3, COD4, COD5, COD6, COD7, COD8, COD9, COD10, COD11, COD12 e CN. A coluna

“CARACTERE” armazena o valor do caractere, as colunas que começam com a sigla “COD” são os códigos do caractere e a coluna “CN” indica se o caractere é uma letra ou um número.

CARA...	COD1	COD2	COD3	COD4	COD5	COD6	COD7	COD8	COD9	COD10	COD11	COD12	CN
A	70	71	70	58	886	886	902	902	58	58	32	0	C
A	71	70	71	58	886	886	902	902	58	10	32	0	C
A	70	70	74	58	58	954	954	906	58	58	224	32	C
A	71	70	75	58	886	954	954	954	58	58	902	32	C
A	70	70	58	58	58	58	954	954	58	13	32	32	C
A	71	75	75	58	58	58	954	954	58	58	32	32	C
A	71	75	58	886	886	954	902	902	58	236	32	0	C
A	70	70	74	58	58	954	954	906	58	58	224	32	C
A	70	71	70	58	886	886	902	902	58	58	32	0	C
A	71	74	74	58	58	58	954	950	58	10	33	0	C
A	71	74	58	58	58	58	950	890	58	13	32	0	C
A	71	70	71	58	886	886	902	902	58	10	32	0	C
A	55	71	58	58	58	950	950	954	58	10	32	32	C
B	58	13	237	32	32	13	44	32	32	49	13	13	C
B	13	13	236	224	224	13	13	32	32	237	13	58	C
B	950	236	32	32	32	236	13	32	32	224	13	0	C
B	58	13	237	32	32	13	44	32	32	49	13	13	C
B	13	13	224	32	32	13	13	224	32	224	13	0	C
B	58	13	13	32	224	13	13	32	32	48	13	0	C
C	58	58	134	32	16	16	16	16	32	134	954	58	C
C	58	58	202	70	16	16	16	16	16	134	202	58	C
C	59	58	186	33	16	16	16	16	32	33	10	58	C

Figura 4.9. Exemplo da tabela “códigos_caracteres”.

Após a codificação do caractere o sistema solicita ao usuário a digitação do valor correspondente a ele, armazenando no banco de dados esse informação e a codificação correspondente (Manfio, 2008).

O processo de reconhecimento executado pelo sistema consiste na procura na tabela pelos códigos encontrados e a verificação qual a qual caractere corresponde esse código.

4.4.2 Pacote **bd_rede_neural**

O pacote `bd_rede_neural` é formado pela classe de conexão com o banco de dados, que é responsável por realizar a conexão com o banco de dados Firebird. A String de conexão está configurada para acessar o banco Firebird denominado `TCC_michel_2009.gdb` utilizando a login `michel` e a senha `tcc_2009` (figura 4.9).

```

public class Conexao {
    public Connection con = null;
    public Statement stm = null;
    public Conexao() {
        try {
            Class.forName("org.firebirdsql.jdbc.FBDriver");
            con =
                DriverManager.getConnection("jdbc:firebirdsql:localhost/
                3050:D:/monografia/software/banco_dados/TCC_michel_2009.
                gdb ", "michel", "tcc2009");
            stm = con.createStatement();
        }
        catch (Exception e) {
            System.out.println("Erro na conexão com o banco " +
                e.getMessage());
        }
    }
}

```

Figura 4.10. Código fonte da classe conexão

4.4.3 Interface do Sistema

A figura 4.11, apresenta a interface do sistema de identificação de placas de carros.

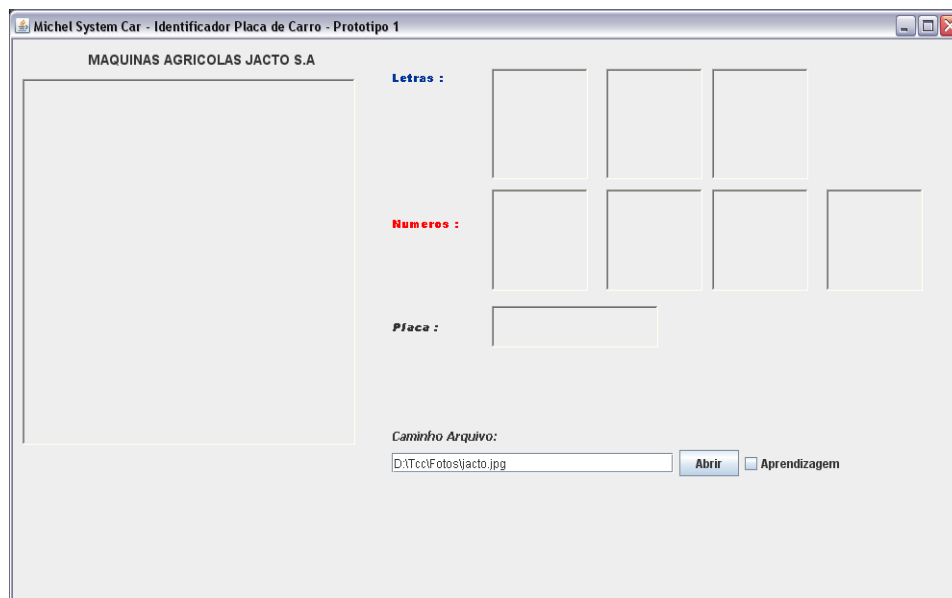


Figura 4.11. Interface do sistema.

O funcionamento deve-se ao fato de buscar as fotos armazenadas no computador clicando na opção **abrir**, o detalhe é que as imagens foram obtidas por uma câmera digital, este protótipo não utiliza uma câmera estilo CCD. Após a imagem do carro, aparecer na tela, o sistema realiza os métodos necessários para encontrar a placa e os caracteres, exibindo o resultado como mostra a figura 4.12.



Figura 4.12. Execução do sistema proposto

5 Conclusões e Perspectiva Futura

Acredita-se que a pesquisa realizada atingiu seus objetivos, pois foram estudadas técnicas de processamento de imagens e implementado o protótipo do sistema de reconhecimento de placas de carro. Com este protótipo deve-se coletar alguns resultados a ser apresentados a empresa Jacto S.A.

Como perspectiva futura sugere-se:

- A utilização de novas técnicas de limiarização e reconhecimento de caracteres, buscando aperfeiçoar o protótipo desenvolvido, ou até mesmo construir um sistema híbrido.
- Como a obtenção de imagens ainda é realizada por uma máquina digital pretendemos utilizar uma câmera automática CCD, para que o processo fique menos dependente da intervenção humana.
- Construir um banco de dados com as características (nome, idade, cargo, etc) e horários de entrada e saída do veículo, construir uma tela para interagir e receber estes parâmetros para que o sistema de identificação realize diferentes processos não apenas na obtenção de dados, mas acrescentando informações do condutor.
- Devem-se buscar alternativas para localização de placa, ou mesmo enviar dados sobre a identificação através de uma rede LAN.

Referências Bibliográficas

BARRETO, G. A., ARAUJO, A. F.R., 1999. *Fast Learning of Robot Trajectories Via Unsupervised Neural Networks*, Proceedings of the 14th IFAC World Congress, Beijing, China 1999.

BARRETO, G. A., ARAUJO, A. F.R., *Storage and recall of complex temporal sequences through a contextually guided self-organizing neural network*, IEEE-INNS-ENNS Intl.Joint Conf. on Neural Networks, July 24-27, Como, Italy, Vol. 3, pp. 207-212, 2000.

BARRETO, G. A., ARAUJO, A. F.R., *Time in Self-Organizing Maps: Na OverView of Models*, International Journal of Computer Research, p.139-171, 2001.

BARROSO, J., RAFAEL, A., DAGLESS, E.A., *Number plate reading using computer vision*. 1997 .

BELLON, O. R. P.. *Segmentação de Imagens em Profundidade*. IN: SIBGRAPI' 1996.

BELVISI, R, FREITAS, R.,. *Um Sistema de Reconhecimento Automático de Placas de Automóveis*. In: Encontro nacional de inteligência artificial, 2. Rio de Janeiro, 1999.

CAMPOS, T., BAMPI, S, SUSIN, A. *Sistema de Identificação de Placas por Processamento Automático de Imagens*. In: VII Workshop IBERCHIP, Montevideo, Uruguay, 2001.

CASTLEMAN, K. R. *Digital Image Processing*, Prentice Hall, 1996.

DIAS, F. G. *Conceitos Fundamentais de SRPLV*. 2007.

DIAS, F. G.; **Melhorias para Sistemas de Reconhecimento da Placa de Licenciamento Veicular**, 2005.

FELAU, I, *Estudo para desenvolvimento de um sistema de extração de características de placas de veículos automotores*, Trabalho de conclusão de curso, Nova Hamburgo, 2008.

FILHO, N., *Processamento Digital de Imagens*. Editora Brasport, 1999.

GONZALEZ, R. C.; WOODS R. E., **Digital Image Processing**, Addison-Wesley Publishing Company, Setembro, 1993.

GONZAGA, A., **Processamento Inicial**. São Paulo. Universidade de São Paulo.

KOVÁCS, Z., *Redes Neurais Artificiais, Fundamentos e Aplicações* –Colegium Cognition, 2^a ed. - São Paulo, 1996.

LEITE, M., *Acessando Bancos de Dados com ferramentas RAD: Aplicações em Visual Basic: Aplicações em Visual Basic*. Brasport, 2007.

MANFIO, F. H., *Controle de acesso a condomínios fechados por meio da identificação de placas de veículos*, 2008.

MARQUES F.O., VIEIRA N.H., **Processamento Digital de Imagens**, Brasport, 1999.

MCCULLOCH, W.S., PITTS, W.A., *A logical calculus of the ideas immament in nervous activity*, Bulletin of Mathematics and Biophysics, Vol. 5, 1943, pp. 115-133.

MINSKY,M., PAPPERT, S., *Perceptrons* – MIT Press, Cambridge, 1969

NEVES, S. C. M.; PELAES, E. G. *Estudo e implementação de técnicas de segmentação de imagens*, Edição N° 02 - outubro / 2001.

POLIDÓRIO, A. M., BORGES,D.L, *Um Método de Reconhecimento Sintático de Caracteres para Identificação de Placas de Veículos*. In: X Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, Campos do Jordão, SP, Outubro 1997.

REILLY, D. REILLY, M., *Java Network Programming and Destributed Computing*. Addison Wesley, Março, 2002.

RODRIGUES, R.J. , *Segmentação e Extração de Características para Reconhecimento Automático de Caracteres – Estudo e Propostas*. (Dissertação de Mestrado). Universidade Federal do Rio de Janeiro, 2003

SANTOS, V. M. F. *Elementos de Imagem e Visão por Computador*. Aula Teórica Robótica Industrial, 2006.

SONKA, M.; HLAVAC, V.; BOYLE, R.. *Image Processing, Analysis, and Machine Vision*, 3. ed. EUA: Thomson Learning, 2007. 829 p.

SOUZA, F. P. C., *Localização e leitura automática de caracteres alfanuméricos – uma aplicação na identificação de veículos*. Universidade Federal Do Rio Grande Do Sul, Curso de Pós-Graduação em Engenharia Elétrica, Porto Alegre, Abril 2000.

Vázquez, N.; Nakano, M.; Pérez-Meana, H.. *Automatic System for Localization na Recognition of Vehicle Plate Numbers*. In: Journal of Applied Research and Technology, p. 63-77.