

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ESCOLA DE ENGENHARIA

Curso de Pós-Graduação em Engenharia Elétrica

CPGEE

**LOCALIZAÇÃO E LEITURA AUTOMÁTICA DE CARACTERES
ALFANUMÉRICOS – UMA APLICAÇÃO NA IDENTIFICAÇÃO
DE VEÍCULOS**

Fernando Peixoto Coelho de Souza

Dissertação para obtenção do título de Mestre em Engenharia

Porto Alegre

Abril 2000

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ESCOLA DE ENGENHARIA

Curso de Pós-Graduação em Engenharia Elétrica

CPGEE

**LOCALIZAÇÃO E LEITURA AUTOMÁTICA DE CARACTERES
ALFANUMÉRICOS – UMA APLICAÇÃO NA IDENTIFICAÇÃO
DE VEÍCULOS**

Fernando Peixoto Coelho de Souza

Engenheiro Eletricista

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia Elétrica – CPGEE, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Processamento de Sinais. Desenvolvida no Laboratório de Processamento de Sinais e Imagens do Departamento de Engenharia Elétrica da UFRGS.

Porto Alegre

Abril 2000

LOCALIZAÇÃO E LEITURA AUTOMÁTICA DE CARACTERES ALFANUMÉRICOS – UMA APLICAÇÃO NA IDENTIFICAÇÃO DE VEÍCULOS

Fernando Peixoto Coelho de Souza

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Altamiro A. Susin, UFRGS

Doutor pelo INPG-Grenoble (F)

Banca Examinadora:

Prof. Dr. Luigi Carro, UFRGS

Doutor pela UFRGS - Porto Alegre

Prof^a. Dr^a. Silvia Delgado Olabarriaga , UFRGS

Doutora pela Universidade de Amsterdam - Holanda

Prof. Dr. Sérgio Bampi, UFRGS

Doutor pela Universidade de Stanford - EUA

Coordenador do CPGEE: _____

Prof. Dr. Renato Machado Brito

Porto Alegre, abril de 2000

Dedico esta dissertação à minha esposa Justine, pela compreensão e dedicação, aos meus filhos Gabriela e Luis Fernando pela alegria que me dão, e aos meus pais João Fernando e Marta e irmãos Gabriela, José, Marília e Nil.

“Não há vento favorável para aquele que não sabe aonde vai”

Sêneca

AGRADECIMENTOS

Ao prof. Dr. Altamiro Amadeu Susin pela orientação, escolha acertada do assunto e confiança depositada, que permitiu minha iniciação e aprendizado nesta maravilhosa área que é o Processamento digital de imagens.

À minha amada esposa que durante esta difícil fase em nossas vidas soube sabiamente compreender e aceitar o esforço dedicado neste trabalho durante todos estes meses.

Aos meus colegas e amigos Eduardo Costa da Motta, Claudio Fernández, Cristiano Brudna, Janor Bastos, Kamel Bensebâa, Ronaldo Hüsemann e outros.

Aos bolsistas Francisco Peixoto Socal pela pronta ajuda prestada no desenvolvimento da interface do software, e Carlo Tomazoni pela paciência e trabalho na aquisição de várias fotos necessárias para o teste do sistema.

Aos colegas de trabalho do LaPSI: Marcelo Negreiros, Thomas Koya, e ao Prof. Adalberto Schuck Júnior.

A todos aqueles que colaboraram de uma forma ou de outra para que este trabalho se desenvolvesse.

À FAPERGS e ao CNPq pelas bolsas de mestrado concedidas.

SUMÁRIO

LISTA DE FIGURAS.....	xi
LISTA DE TABELAS E GRÁFICOS.....	xiii
RESUMO.....	xiv
ABSTRACT.....	xvi

CAPÍTULO 1

JUSTIFICATIVAS DO PROBLEMA DA PESQUISA E METODOLOGIA

1.0 INTRODUÇÃO.....	1
1.1 CONTEXTO DO ESTUDO.....	2
1.2 OBJETIVO.....	3
1.3 PROBLEMÁTICA DA SEGMENTAÇÃO.....	4
1.4 ESCOLHA DO ALGORITMO.....	5
1.5 METODOLOGIA.....	6
1.6 ORGANIZAÇÃO DO TRABALHO.....	7

CAPÍTULO 2

CONCEITOS FUNDAMENTAIS

2.0 INTRODUÇÃO.....	9
2.1 DEFINIÇÃO DE IMAGEM E IMAGEM DIGITAL.....	9
2.2 CARACTERÍSTICAS DAS IMAGENS DIGITAIS.....	10
2.2.0 Imagem binária.....	10
2.2.1 Imagem em tons de cinza.....	10
2.2.2 Imagem colorida.....	11
2.3 ARMAZENAMENTO DIGITAL DE IMAGENS.....	13
2.4 FORMATOS PARA ARMAZENAMENTO.....	14
2.5 ELEMENTOS DE UM SISTEMA DE PROCESSAMENTO DIGITAL DE IMAGENS.....	15
2.6 PROCESSAMENTO E ANÁLISE DE IMAGENS.....	18

CAPÍTULO 3

PROCESSAMENTO DE IMAGEM

3.0 INTRODUÇÃO	20
3.1 REALCE DE IMAGENS	20
3.1.0 Técnicas de realce por modificação no histograma.....	20
3.1.1 Definição do histograma de uma imagem	21
3.1.2 Alteração de contraste por modificação no histograma	21

CAPÍTULO 4

ANÁLISE DE IMAGEM

4.0 INTRODUÇÃO	25
4.1 SEGMENTAÇÃO DE IMAGENS	25
4.1.0 Binarização com limiar global:	26
4.1.1 Binarização local adaptativa:.....	27
4.1.2 Binarização Niblack	28
4.2 REPRESENTAÇÃO E DESCRIÇÃO DE OBJETOS	30
4.2.0 Representação por projeção vertical e horizontal.....	31

CAPÍTULO 5

RECONHECIMENTO E INTERPRETAÇÃO DE IMAGENS

5.0 INTRODUÇÃO	32
5.1 ELEMENTOS DE ANÁLISE DE IMAGEM	33
5.2 PROCESSAMENTO DE ALTO NÍVEL	34
5.3 Redes Neurais	35
5.3.1 Modelo de um neurônio.....	36
5.3.2 Arquitetura da rede	37
5.3.3 Operação da rede	38
5.3.4 Processo de aprendizado.....	39
5.3.5 Redes neurais multilayer <i>feedforward</i>	39

CAPÍTULO 6

SISTEMA DE IDENTIFICAÇÃO AUTOMÁTICA DE VEÍCULOS

6.0 INTRODUÇÃO	45
6.1 INTRODUÇÃO AOS SISTEMAS DE IDENTIFICAÇÃO AUTOMÁTICA DE VEÍCULOS	45
6.2 APLICAÇÕES	46
6.3 ALGORITMOS DE LOCALIZAÇÃO E SEGMENTAÇÃO DOS SISTEMAS ESTUDADOS	49
6.3.1 Identificação da placa através de análise por variação tonal	49
6.3.2 Identificação da placa pela procura de elementos em uma imagem binarizada	51
6.4 O SISTEMA DESENVOLVIDO	52
6.4.1 Conversão de uma imagem colorida para monocromática.....	54
6.4.2 Análise por variação tonal	55
6.4.3 Análise dos dígitos	59
6.4.3.1 Binarização Niblack modificada	59
6.4.3.2 Localização dos dígitos	60
6.4.3.3 Verificação de grupos válidos	62
6.4.4 Verificação do número de caracteres e padronização dimensional.....	63
6.4.5 Processamento da região da placa	65
6.4.6 Extração dos caracteres encontrados	66
6.4.7 Redimensionamento dos caracteres selecionados	69
6.4.8 Reconhecimento dos caracteres.....	70

CAPÍTULO 7

RESULTADOS

7.0 INTRODUÇÃO	72
7.1 ANÁLISE POR VARIAÇÃO TONAL	72
7.2 ANÁLISE POR DÍGITOS.....	75
7.3 REDES NEURAIIS	79
7.4 RESULTADOS GERAIS E COMPARATIVOS	80

CAPÍTULO 8

CONCLUSÃO

8.0 CONCLUSÃO	81
8.1 PERSPECTIVAS DE TRABALHOS FUTUROS	82
APÊNDICE A – SOFTWARE	84
APÊNDICE B - EQUIPAMENTOS.....	86
BIBLIOGRAFIA	87

LISTA DE FIGURAS

Fig. 1.1– Exemplos de imagens pertencentes ao banco de testes utilizado	4
Fig. 1.2– Detalhe de diferentes placas das imagens escolhidas	5
Fig. 2.1 – Função $\delta(x,y)$	9
Fig. 2.2 – Anatomia do olho humano	11
Fig. 2.3 – Sensibilidade espectral dos três tipos de cones	12
Fig. 2.4 – Formação da imagem colorida	12
Fig. 2.5 – Elementos de um sistema de processamento e análise de imagem	17
Fig. 3.1 – Imagem e histograma	21
Fig. 3.2 – Imagem com baixo contraste e seu histograma	22
Fig. 3.3 – Imagem realçada e seu histograma	22
Fig. 3.4 – Variação do contrast stretching	23
Fig. 4.1 – Histograma bimodal	26
Fig. 4.2 – Imagem com histograma bimodal	27
Fig. 4.3 – Resultado de uma binarização por <i>threshold</i> global.....	27
Fig. 4.4– Imagem com iluminação não uniforme e histograma	28
Fig. 4.5 – Resultado de uma binarização por <i>threshold</i> global.....	28
Fig. 4.6 – Resultado da binarização Niblack para diferentes janelas de amostragem	30
Fig. 4.7 – Projeções horizontais e verticais.....	31
Fig. 5.1 – Elementos de análise de imagem.....	33
Fig. 5.2 – Modelo não linear de um neurônio.....	37
Fig. 5.3 – Topologia <i>feedforward</i> e recorrente	38
Fig. 5.4 – Topologia da rede <i>feedforward</i>	40
Fig. 5.5 – Fase de propagação.....	43
Fig. 5.6 – Fase de adaptação	44
Fig. 6.1 – Configuração típica de um SIAV	46
Fig. 6.2 – Imagem original e intensidades das linhas A e B.....	49
Fig. 6.3 – Etapas do sistema	53
Fig. 6.4 – Imagem original e saída do operador Sobel	56
Fig. 6.5 – Detalhe da imagem original e saída do operador	56
Fig. 6.6 – Sentido de varredura.....	56

Fig. 6.7 – Resultado do filtro Sobel e detalhe da marcação dos picos.....	57
Fig. 6.8 – Resultado do filtro Sobel e da etapa de marcação e eliminação de picos	57
Fig. 6.9 – Resultado da etapa de marcação e eliminação de picos	58
Fig. 6.10 – Resultado final do algoritmo de análise por variação tonal e detalhe	58
Fig. 6.11 – Fases da análise de dígitos.....	59
Fig. 6.12 – Resultado do algoritmo de binarização Niblack.....	60
Fig. 6.13 – Janela de varredura	60
Fig. 6.14 – Algoritmo de localização de dígitos	60
Fig. 6.15 – Algoritmo de localização de dígitos – estágio 1	61
Fig. 6.16 – Algoritmo de localização de dígitos – estágio 2.....	61
Fig. 6.17 – Algoritmo de localização de dígitos – estágio 3.....	61
Fig. 6.18 – Ponto de referência (âncora).....	62
Fig. 6.19 – 6 caracteres encontrados.....	63
Fig. 6.20 – Intervalos (A>B).....	64
Fig. 6.21 – Novo caracter (NC)	64
Fig. 6.22 – Padronização dimensional	65
Fig. 6.23 – Área de atuação do contrast stretching	66
Fig. 6.24 – Imagem original e histograma compactado.....	66
Fig. 6.25 – Resultado do contrast stretch e histograma expandido.....	66
Fig. 6.26 – Imagem original e resultado do contrast stretching.....	66
Fig. 6.27 – Orientação da procura dos pixels vizinhos	67
Fig. 6.28 – Etapas da segmentação por similaridade	67
Fig. 6.29 – Caracteres inscritos e suas dimensões	68
Fig. 6.30 – Caracteres e ruídos provenientes da etapa de binarização.....	68
Fig. 6.31 – Resultado da análise dos elementos encontrados	68
Fig. 6.32 – Dimensões dos caracteres	69
Fig. 6.33 – Caracteres selecionados sem dimensão fixa.....	69
Fig. 6.34 – Caracteres redimensionados para matriz 15x15 pixels.....	69
Fig. 6.35 – Caracter I (15x15 pixels)	70
Fig. 6.36 – Redes neurais para letras e números.....	70
Fig. 7.1 - Operador Sobel.....	73
Fig. 7.2 - Marcação dos intervalos.....	73
Fig. 7.3 - Resultado final	73

Fig. 7.4 - Operador Sobel.....	74
Fig. 7.5 - Marcação dos intervalos.....	74
Fig. 7.6 - Resultado final	74
Fig. 7.7 – Etapas do sistema	75
Fig. 7.8 – Etapas do sistema	76
Fig. 7.9 – Etapas do sistema	77
Fig. 7.10 – Etapas do sistema	78
Fig. 7.11 – Identificação correta dos sete caracteres	79
Fig. 7.12 – Identificação correta de seis caracteres	79
Fig. A1 – Interface do software	84
Fig. A2 – Janela de configuração.....	85

LISTA DE TABELAS E GRÁFICOS

Tabela 1 – Coeficientes do operador Sobel	55
Tabela 2 – Associação dos neurônios da camada de saída	71
Gráfico 1 – Resultados comparativos das redes neurais.....	79
Gráfico 2 – Resultados comparativos dos sistemas	80

RESUMO

Sistemas de visão artificial são cada vez mais usados para auxiliar seres humanos em diferentes tarefas. Estes sistemas são capazes de reconhecer padrões previamente ensinados em uma imagem complexa. A leitura automática é uma das mais atraentes tarefas nesta área [1], sendo que uma máquina com esta capacidade pode reconhecer objetos que possuam caracteres em sua identificação.

Na área de trânsito, a identificação de veículos através da leitura de sua placa de licença vem conquistando cada vez mais espaço. No início dos anos cinquenta, este conceito era usado para estudar o tempo de duração de viagens entre origem e destino. Os primeiros métodos utilizados eram baseados em observadores que anotavam as placas dos veículos e os tempos correspondentes em um papel ou fita gravada. As placas eram manualmente comparadas mais tarde, e os tempos de viagem calculados [2]. O crescente avanço tecnológico tem aumentado substancialmente a precisão e facilidade desta técnica permitindo sua utilização na identificação de veículos infratores e em situação irregular, e no controle de pedágios e estacionamentos pagos.

Este trabalho envolve o estudo de diversas técnicas de processamento e análise de imagem culminando no desenvolvimento de um sistema capaz de localizar e reconhecer os caracteres contidos numa placa de licença de um veículo. A imagem é previamente analisada por um algoritmo de procura por variações tonais padronizadas de maneira a restringir a área de análise do algoritmo principal do sistema. Este, por sua vez, binariza a imagem através de um algoritmo adaptativo e busca elementos que possuam dimensões próximas às dimensões esperadas dos caracteres da placa. O sistema busca encontrar uma seqüência de caracteres de dimensões aproximadamente iguais e para isso, varia um valor de limiar no processo de binarização conferindo maior robustez ao algoritmo. Uma vez encontrado um grupo de dígitos que satisfaçam alguns critérios pré-definidos, os caracteres são redimensionados e apresentados a duas redes neurais, uma para as letras e outra para os números.

O sistema desenvolvido foi testado com um conjunto de 500 imagens com diferentes características e tipos de distribuição de luz sobre a superfície da placa. O sistema apresentou 97,7 % de sucesso na localização das placas e 91,3 % na extração correta dos caracteres nela contidos. Estes valores são aproximadamente 20 % superiores aos resultados apresentados por um sistema comercial israelense [3] também testado com o mesmo banco de imagens. Os resultados obtidos com as rede neurais no reconhecimento dos caracteres (81,3 %), e no reconhecimento das placas (37,5 %), não foram satisfatórios, sendo necessário um estudo mais aprofundado nesta área.

ABSTRACT

Artificial vision systems are used more and more to aid human beings in different tasks. These systems are capable of recognizing patterns previously trained in a complex image. The automatic reading is one of the most attractive tasks in this area [1], and a machine with this capacity can recognize objects that possess characters in its identification.

In the traffic area, the identification of vehicles through the reading of its license plate has been conquering more space. In the beginning of the fifties, this concept was used to study the duration of trips between origin and destiny. The first methods used observers that wrote down the plates of the vehicles and the corresponding times in a paper or recorded tapes. The plates were later manually compared, and the time of the trip was calculated [2]. The technological progress has been increasing the precision and easiness of this technique, allowing its use in the offenders vehicles' identification in irregular situation, and in the control of tolls and parkings.

This work involves the study of several processing techniques and image analysis culminating in the development of a system capable to locate and to recognize the characters contained in a vehicle license plate. The image is analyzed previously by a search algorithm by standardized tonal variations in a way to restrict the area of analysis of the main algorithm of the system. This, for its time, binarizes the image through an adaptative algorithm, looking for elements that possess close dimensions to the expected dimensions of the plate's characters. The system searches for a sequence of characters of approximately same dimensions and for that, it varies a threshold value in the binarization process conferring larger robustness to the algorithm. Once a group of digits that satisfy some previously defined approaches is found, the characters are redimensioned and presented to two neural networks, one for the letters and another for the numbers.

The developed system was tested with a group of 500 images with different characteristics and types of light distribution on the surface of the plate. The system presented 97,7% of success in the location of the plates and 91,3% in the correct extraction of the characters contained in it. These values are approximately 20% superior to the results also presented by an Israeli commercial system [3] tested with the same image bank. The results obtained with neural networks in the recognition of the characters were successful in 81,3%, and in the recognition of the plates 37,5%. These numbers were not satisfactory, being necessary a more deep study in this area.

CAPÍTULO 1

JUSTIFICATIVAS DO PROBLEMA DA PESQUISA E METODOLOGIA

1.0 INTRODUÇÃO

A visão e a audição são os dois principais meios pelos quais os seres humanos percebem o mundo exterior [4]. O processamento digital de imagens – a manipulação de imagens por computador – é relativamente recente em relação à fascinação humana por estímulos visuais [5]. Uma das primeiras aplicações de que se tem notícia foi a utilização de técnicas de processamento de imagem na melhora da qualidade visual de fotos de jornal digitalizadas. As imagens eram enviadas, através de um cabo submarino, de Londres à Nova York, e então processadas para visualização. A utilização do sistema *Bartlane* de transmissão de fotos no início dos anos 20, reduziu o tempo gasto nesta tarefa, que era de mais de uma semana, para menos de três horas [6]. Em 1964, fotos da lua enviadas pela missão Ranger 7 foram processadas com o objetivo de corrigir vários tipos de distorções inerentes à câmera utilizada. As técnicas desenvolvidas nesta época serviram de base para o desenvolvimento de outros algoritmos que foram utilizados nas missões espaciais seguintes.

De 1964 até hoje, o processamento digital de imagem vem numa crescente expansão. Com o avanço tecnológico a passos largos, e a diminuição dos custos relativos aos computadores e seus periféricos, ele vem conquistando cada vez mais espaço. Além das aplicações relativas às missões espaciais, as técnicas de processamento digital de imagem são hoje empregadas na resolução de uma gama de diferentes problemas. Embora não relacionados, estes problemas normalmente necessitam de métodos capazes de realçar informações para interpretação humana ou processar dados de uma cena para percepção automática.

Como em outros campos multidisciplinares, o processamento de imagem é composto de diversas áreas: óptica, eletrônica, matemática, fotografia e computação. Suas potenciais aplicações são ilimitadas, a robótica, a automação, a astronomia, a biologia, a agricultura, a antropologia e a física são apenas algumas delas. Todavia, a aplicação de maior impacto em nossas vidas é na medicina [4]. A tomografia

computadorizada é usada rotineiramente em muitos diagnósticos clínicos, como, por exemplo, na detecção e identificação de tumores no cérebro. Outras aplicações médicas incluem realce de imagens de raio X, contagem de reticulócitos, detecção de células cancerígenas, etc.

Segundo Gonzalez [6], o processamento digital de imagens pode ser dividido em duas principais aplicações: (1) melhora do aspecto visual de uma imagem para interpretação humana, e (2) processamento dos dados de uma cena para percepção automática de máquinas. Esta última envolve técnicas de inteligência artificial, e tem como objetivo dotar uma máquina da capacidade de executar tarefas que exijam um certo grau de inteligência, possibilitando que o sistema interprete os elementos de interesse pertencentes a cena.

O desenvolvimento de um sistema automático de análise de imagem, alvo deste trabalho, constitui uma das mais difíceis tarefas dentro do campo de processamento de imagem e será melhor explicado no decorrer desta dissertação.

1.1 CONTEXTO DO ESTUDO

Um sistema automático de análise de imagem, também denominado sistema de visão artificial, tem como objetivo adquirir, processar, analisar e interpretar um determinado tipo de imagem para uma determinada aplicação. Geralmente, estes sistemas possuem aplicações bem específicas, permitindo que o sistema apresente um alto desempenho para uma determinada classe de imagem.

A identificação de veículos através da leitura da placa de licença era usada já nos anos cinquenta para estudar o tempo de duração de viagens entre origem e destino. Os métodos iniciais eram baseados em observadores que anotavam as placas dos veículos e os tempos correspondentes em um papel ou fita gravada. As placas eram manualmente comparadas mais tarde e os tempos de viagem calculados [2].

A tecnologia de vídeo tem desempenhado um papel fundamental para o levantamento de parâmetros relacionados à engenharia de tráfego. A medição e o monitoramento de várias características de viagens e fluxo de tráfego urbano por meio da tecnologia de vídeo está sendo usada cada vez mais para o planejamento de transporte e gerenciamento do tráfego [7]. A possibilidade de se determinar padrões de

movimento, pela comparação de pares de placas de licença ao longo de uma malha rodoviária, levou à primeira implementação de um sistema de identificação automática de veículos em 1990 na Inglaterra. A praticidade deste tipo de sistema foi confirmada em uma série extensiva de tentativas conduzidas nos Estados Unidos, em 1993, pela *Volpe National Transportation Systems Center (VMTSC)* e seus associados [7].

Além do controle e monitoramento do tráfego, um sistema de identificação automática de veículos possui outras aplicações, como:

- Reconhecimento de veículos em situação irregular;
- Controle de pedágios e estacionamentos;
- Identificação de veículos infratores;

1.2 OBJETIVO

O objetivo deste trabalho é o desenvolvimento de um sistema de visão artificial capaz de localizar e reconhecer o conteúdo de uma placa de licença de um veículo, sem necessitar de qualquer intervenção humana. Isto se dará através da utilização conjunta de técnicas de processamento e análise de imagem, para a localização e segmentação dos caracteres respectivamente, e técnicas de inteligência artificial para leitura dos mesmos.

O sistema será projetado para trabalhar com imagens que possuam as seguintes características:

- Placas com 7 caracteres (três letras e quatro números);
- Fundo claro e caracteres escuros;
- Imagens da frente e traseira dos veículos;
- Distribuição de luz homogênea e heterogênea sobre a superfície da placa;

O banco de imagens que será utilizado para avaliar o desempenho do sistema deverá ser composto de imagens reais, adquiridas em situações cotidianas como estacionamentos e trânsito livre. A figura 1.1 exemplifica o que foi dito anteriormente.



Fig. 1.1– Exemplos de imagens pertencentes ao banco de testes utilizado

1.3 PROBLEMÁTICA DA SEGMENTAÇÃO

O projeto de um sistema de visão artificial consiste na escolha adequada de uma seqüência de algoritmos de processamento, análise e interpretação de imagens, capazes de oferecer uma solução adequada para uma determinada aplicação. Dentre os diversos algoritmos utilizados nos diversos estágios que compõem um sistema destes, o algoritmo responsável pela segmentação automática dos caracteres é o mais importante. Segundo Gonzalez [6], a segmentação automática trata de uma das mais difíceis tarefas em processamento de imagem. Esta etapa é responsável por um eventual sucesso ou falha de um sistema de visão artificial, devendo merecer atenção especial em seu projeto.

Como será visto no decorrer deste trabalho, a escolha do algoritmo de segmentação dos caracteres influencia duas etapas fundamentais do sistema: a localização da placa e a extração dos caracteres do restante da imagem.

A utilização de imagens do trânsito cotidiano, adquiridas sem controle de iluminação, bem como a possibilidade de existirem elementos não previstos presentes na imagem faz desta aplicação em particular um verdadeiro desafio.

1.4 ESCOLHA DO ALGORITMO

A escolha apropriada de um algoritmo eficiente para segmentação automática em uma determinada aplicação pode ser feita em três passos. O primeiro passo, consiste em uma escolha adequada das imagens que serão utilizadas para avaliação de desempenho do algoritmo. Este passo é essencial para garantir robustez ao sistema, uma vez que estas imagens devem retratar as mais variadas situações de trânsito e condições de iluminação das placas dos veículos (Fig. 1.2).



Fig. 1.2– Detalhe de diferentes placas das imagens escolhidas

O segundo passo consiste em uma análise minuciosa das imagens escolhidas em busca de características peculiares que sejam capazes de distinguir o objeto de interesse dos demais elementos presentes na cena. Este procedimento dará uma idéia dos problemas que deverão ser enfrentados, e fornecerá subsídios para a escolha de um algoritmo adequado para a aplicação em questão.

Como último passo, tendo em mente os obstáculos oferecidos pela aplicação, é necessário uma análise sobre os diversos algoritmos de segmentação existentes. A escolha do algoritmo apropriado para utilização nesta aplicação deverá levar em conta os seguintes fatores:

Robustez – o algoritmo deverá ser capaz de segmentar satisfatoriamente caracteres submetidos a diferentes condições de iluminação com a menor distorção possível dos mesmos.

Velocidade – o algoritmo deverá ser suficientemente rápido para permitir que o sistema opere em aplicações reais.

Precisão – o algoritmo deverá ser preciso na segmentação dos caracteres uma vez que dele depende a localização da placa e a extração correta dos caracteres..

Após uma análise dos algoritmos encontrados na literatura, chegou-se à conclusão de que a binarização local adaptativa *Niblack* [8] corresponde às exigências da aplicação. Este algoritmo será visto com detalhes mais adiante.

1.5 METODOLOGIA

Para localizar e interpretar o conteúdo de uma placa de um veículo uma série de estágios são necessários. Após a captura da imagem através da utilização de algum dispositivo apropriado (câmera CCD – *Charge Couple Device*), e sua digitalização através de uma placa de aquisição de vídeo (*frame grabber*), a imagem digital é processada, analisada e interpretada segundo os estágios abaixo:

1- O primeiro estágio do sistema consiste na conversão da imagem adquirida para 256 tons de cinza, uma vez que a imagem capturada pela placa de aquisição é colorida e os algoritmos empregados não fazem uso desta informação

2- O segundo estágio consiste de uma análise na imagem a procura de um padrão de variação tonal conhecido – “assinatura” da placa. Este procedimento é intuitivo, visto que se baseia na existência de caracteres contrastando com um fundo, uma característica obrigatória em qualquer placa de licença. Os elementos contrastantes podem ser localizados através de uma análise por discontinuidades. No caso de alguma placa ser detectada neste estágio, sua localização é armazenada para processamento no próximo estágio.

3- O terceiro estágio é o mais importante de todos e consiste de três etapas. Na primeira tem-se uma binarização local adaptativa sobre a imagem inteira ou sobre o resultado conseguido no estágio anterior. Na segunda etapa um algoritmo de localização de dígitos é utilizado sobre a imagem resultante da etapa de binarização, localizando elementos que possuam dimensões compatíveis com um padrão esperado. E finalmente, na terceira etapa, tem-se a utilização de um algoritmo de análise dos dígitos encontrados. Nesta etapa é verificada a existência de algum grupo de dígitos com as

propriedades espaciais de uma placa de licença; distâncias relativas entre caracteres e dimensões dos mesmos.

4- O quarto estágio consiste no realce da área da placa na imagem original através de um algoritmo de *contrast stretching* [4]. Este estágio permite uma melhora na relação do contraste entre os caracteres e o fundo da placa, propiciando uma melhor extração dos mesmos.

5- O quinto estágio é uma repetição do segundo estágio, porém apenas sobre a imagem realçada. Os três passos: binarização, localização e análise dos dígitos são executados novamente sobre a região da imagem que contém a placa realçada, até que seja encontrado o número de caracteres esperados.

6- O sexto estágio consiste na extração dos caracteres contidos na área da placa. Através da utilização de um algoritmo de agrupamento por similaridade e um conjunto de heurísticas os caracteres são extraídos de uma imagem binarizada.

7- O sétimo estágio consiste no redimensionamento dos caracteres encontrados para utilização no próximo estágio.

8- O oitavo estágio consiste na apresentação dos caracteres encontrados à rede neural para seu reconhecimento. Os três primeiros caracteres são apresentados à rede treinada para reconhecimento de letras e os caracteres restantes são apresentados à rede treinada para reconhecimento de números.

1.6 ORGANIZAÇÃO DO TRABALHO

O presente trabalho é formado por oito capítulos, além da bibliografia utilizada e dos anexos contendo a interface do *software* desenvolvido e os equipamentos utilizados.

O capítulo 2 apresenta alguns conceitos básicos e definições fundamentais para o entendimento deste trabalho. Nele são vistos os conceitos de imagem e imagem digital, bem como os tipos de imagem com relação as cores, seu armazenamento e seus principais formatos de gravação. Além disso, são apresentados os elementos que compõem um sistema de processamento digital de imagem, e as diferenças entre processamento e análise de imagem.

Nos capítulos 3 e 4 são apresentadas técnicas de processamento e análise de imagem respectivamente, utilizadas no desenvolvimento do sistema proposto neste trabalho. Ambos os capítulos apresentam estas técnicas de uma forma geral permitindo ao leitor generalizar sua utilização.

No capítulo 5 tem-se uma introdução a utilização de redes neurais como técnica de interpretação e reconhecimento de imagens. Neste capítulo uma ênfase maior é dada a rede neural *feedforward* com algoritmo de treinamento *error backpropagation* por ser utilizada no reconhecimento dos caracteres extraídos da imagem.

No capítulo 6 é apresentado o conceito e as aplicações de sistemas de identificação de veículos através da leitura da placa de licença. Ainda neste capítulo são apresentados os dois principais métodos encontrados na literatura pesquisada para solucionar este tipo de problema, e em detalhes é apresentado o sistema proposto.

No capítulo 7 são apresentados alguns resultados obtidos com o sistema desenvolvido, mostrando cada etapa do sistema para algumas imagens de exemplo. Os resultados gerais para o banco de imagens utilizados são apresentados ao final do capítulo juntamente com os resultados obtidos por um *software* comercial demonstrativo israelense para o mesmo banco.

E finalmente, no capítulo 8 é apresentada a conclusão do trabalho e as perspectivas de melhoramentos possíveis.

CAPÍTULO 2

CONCEITOS FUNDAMENTAIS

2.0 INTRODUÇÃO

Neste capítulo serão vistos alguns conceitos fundamentais para o entendimento do que será apresentado a seguir. Inicialmente, será definido o que é imagem digital, suas características e formatos mais comuns de armazenamento. Serão apresentados os elementos que compõem um sistema de processamento de imagem, e uma definição para os termos processamento e análise de imagens.

2.1 DEFINIÇÃO DE IMAGEM E IMAGEM DIGITAL

A palavra imagem é atribuída a uma função bidimensional $f(x,y)$ contínua, onde para qualquer par (x,y) existe um valor f proporcional a intensidade do brilho da imagem naquele ponto. As coordenadas espaciais (x,y) localizam qualquer ponto pertencente a imagem em questão.

Como os computadores não manipulam dados analógicos, é necessário converter uma imagem contínua em sua forma digital. Teoricamente, isso pode ser feito através da multiplicação da imagem contínua $f(x,y)$ por uma função delta de Dirac bidimensional, $\delta(x,y)$ (Fig. 2.1).

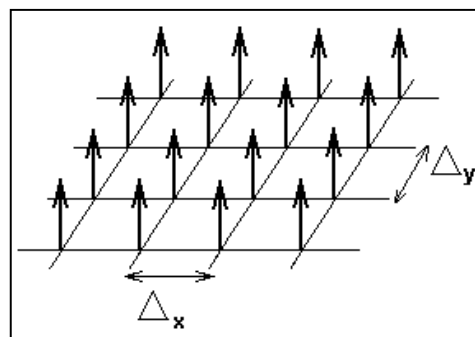


Fig. 2.1 – Função $\delta(x,y)$

O termo imagem digital refere-se a uma imagem que pode ser discretizada quanto as suas coordenadas espaciais e quanto a intensidade de seu brilho. Logo, uma imagem digital pode ser considerada uma matriz cujos índices das linhas e colunas identificam um ponto dentro da imagem e o correspondem ao valor do elemento da matriz que

identifica o nível de intensidade luminosa naquele ponto [6]. Os elementos desta matriz são chamados de *picture elements* mais popularmente conhecidos como *pixels*.

2.2 CARACTERÍSTICAS DAS IMAGENS DIGITAIS

As imagens digitais são classificadas em relação ao número de cores. Esta diferença na representação de uma imagem digital é decorrência do número de bits utilizado pelo dispositivo de aquisição na captura da mesma, sendo normalmente passível de ser selecionado pelo usuário.

2.2.0 Imagem binária

Uma imagem binária, geralmente em preto e branco, é normalmente utilizada para distinguir dois tipos de elementos, o fundo e os objetos de interesse. A maioria dos algoritmos de análise de imagem operam sobre este tipo, retirando informações relacionadas à geometria dos objetos, como área, perímetro, ou quantificando estes elementos. Este tipo de imagem necessita de apenas 1 bit por *pixel* para ser representada, 0 (preto) e 1 (branco), e geralmente é resultado de um processo de segmentação sobre uma imagem com um maior número de cores.

2.2.1 Imagem em tons de cinza

Uma imagem em tons de cinza contém a intensidade de seus *pixels* representada por valores inteiros positivos, dentro de uma escala que varia do preto ao branco, por este motivo esta classe de imagem também é conhecida como imagem monocromática.

Normalmente, a intensidade em um ponto qualquer pode variar entre 256 tons de cinza, ou seja, para cada ponto da imagem podem existir 256 valores diferentes variando do 0 (preto) ao 255 (branco). Neste tipo de imagem são necessários 8 bits por *pixel* para representar os 256 diferentes tons possíveis. Para que se possa ter uma idéia, a representação de uma imagem pequena, de 320x240 *pixels*, requer 76800 bytes.

2.2.2 Imagem colorida

A mais familiar forma de imagem multi-espectral é a colorida. A retina do olho humano é coberta por células foto-receptoras (Fig. 2.2) 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 [5].

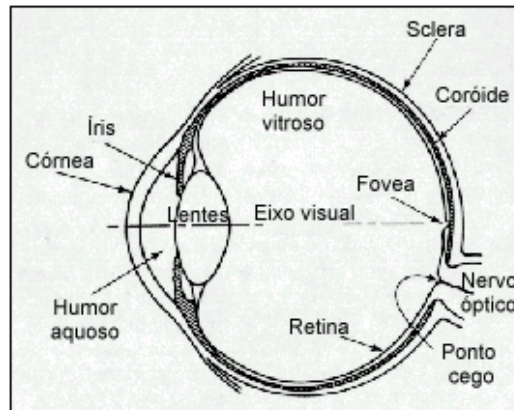


Fig. 2.2 – Anatomia do olho humano (adaptado de Lim [4] pg.424)

Existem dois tipos de células foto-receptoras com diferentes funções: os cones e os bastonetes. Os cones estão localizados basicamente sobre a porção central da retina, chamada *fovea*. Encontrados em torno de 6 a 7 milhões por olho, são extremamente sensíveis a cores. Os seres humanos são capazes de distinguir mínimos detalhes graças a estas células, pois cada uma possui sua própria terminação nervosa.

Em maior número, cerca de 75 a 150 milhões de células por olho, os bastonetes localizam-se sobre toda a superfície da retina. Diferentemente dos cones, várias células são ligadas ao mesmo nervo óptico diminuindo a capacidade de detalhamento delas. Os bastonetes são utilizados pelo sistema de visão humano para dar uma visão geral da cena. Apesar de não serem sensíveis a cores, são responsáveis pela nossa visão noturna graças a sua sensibilidade a baixos níveis de iluminação.

Os cones são divididos em três categorias conforme o processo foto químico que utilizam para a conversão de luz em impulsos nervosos, dividindo a porção visível do espectro eletromagnético em três bandas: vermelho, verde e azul. Por esta razão, estas

três cores são denominadas de cores primárias [5]. A figura 2.3 mostra a sensibilidade espectral dos três tipos de cones existentes no olho humano.

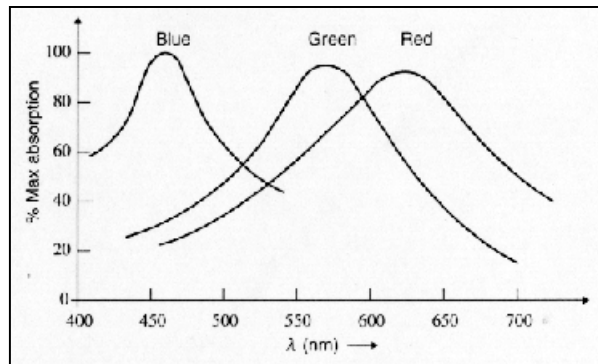


Fig. 2.3 – Sensibilidade espectral dos três tipos de cones

Uma imagem digital colorida pode ser vista como a sobreposição de três imagens monocromáticas [5], cada uma delas representando as intensidades das cores vermelho (*red*), verde (*green*) e azul (*blue*) que compõem a imagem original. Em outras palavras, um *pixel* em uma imagem colorida é o resultado da sobreposição das intensidades dos *pixels* espacialmente alinhados das imagens monocromáticas (Fig. 2.4).

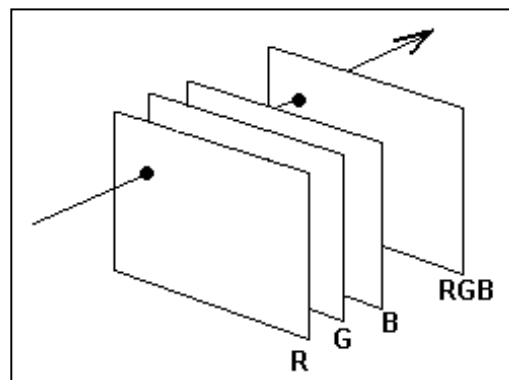


Fig. 2.4 – Formação da imagem colorida

Normalmente, os valores dos *pixels* de cada uma das imagens correspondente a cada uma das três cores primárias, vermelho, verde e azul são representados por inteiros entre 0 e 255. Por este motivo são necessários 24 bits para sua representação, 8 bits para cada uma das três componentes.

Para que haja um *pixel* preto na imagem colorida é necessário que cada um dos *pixels* correspondentes nas imagens monocromáticas sejam iguais a zero, neste caso tem-se a tríade (0,0,0). Para conseguir-se um *pixel* vermelho na imagem RGB precisa-se

que a componente vermelha da tríade seja igual a 255 e as restantes iguais a zero (255,0,0), e assim por diante.

2.3 ARMAZENAMENTO DIGITAL DE IMAGENS

Uma vez que as imagens tenham sido digitalizadas, um procedimento comum é armazená-las em algum dispositivo capacitado para isso. O espaço necessário para o armazenamento de imagens digitais é um fator crítico. Uma página de texto de um livro digitalizado, utilizando um byte por ponto e uma resolução de 300 pontos por polegada, necessita de 4,86 Megabytes em disco. Um livro de 300 páginas digitalizado com as mesmas características acima, necessitaria de 1,5 Gigabytes de espaço para seu armazenamento.

O objetivo da compressão de imagens é reduzir a quantidade de dados necessários para a representação da informação presente na imagem. Os métodos de compressão podem ser divididos em duas categorias básicas: compressão com perdas e sem perdas.

Métodos de compressão com perdas normalmente exploram deficiências do nosso sistema de visão. O olho humano é muito mais sensível a mudanças de brilho do que a mudanças de coloração. Pode-se dizer que é possível suprimir algumas informações de coloração sem que haja perda perceptível na imagem. De fato, todos os maiores padrões de vídeo, incluindo os sistemas NTSC, PAL e SECAM, vêm explorando há anos esta pequena deficiência do olho humano. Cada um destes padrões transmite as informações de luminosidade a uma máxima largura de banda e as informações de coloração a uma largura de banda reduzida.

A maioria dos métodos de compressão sem perdas são baseados na remoção de códigos redundantes que estão presentes em uma imagem utilizando-se de uma codificação com comprimento variável. Dentro de numerosas aplicações a compressão de dados sem perdas se faz necessária. Na área médica ou em documentos de negócios, a perda de informação é proibida por lei. Na radiografia digital, por exemplo, uma perda de informação pode comprometer diagnósticos precisos, e em imagens geradas por satélites o uso e o custo destas informações tornam a compressão com perdas indesejável.

2.4 FORMATOS PARA ARMAZENAMENTO

Atualmente, existem diversos formatos de armazenamento disponíveis pelos *softwares* de manipulação de imagens digitais. Os mais conhecidos são: BMP, JPEG, PCX, TIF, GIF e TGA. Como foge do escopo deste trabalho um estudo detalhado dos formatos citados, explicar-se-ão sucintamente dois formatos em particular: o JPEG e o BMP.

A sigla JPEG, *Joint Photographic Experts Group*, refere-se a um padrão para compressão de imagens fotográficas especificado por um comitê internacional formado pela ISO e pela CCITT, órgãos que definem os padrões para telefonia, rádio e televisão.

O JPEG é, na verdade, uma família de técnicas de compressão onde são suprimidas metodicamente algumas informações visuais insignificantes. Estas informações incluem componentes visuais de alta frequência, que são menos importantes do que alguns componentes visuais de baixa frequência. A imagem recuperada não será necessariamente idêntica à imagem original, o que caracteriza um método de compressão de dados com perda. O método de compactação JPEG permite que o usuário, ou aplicativo, ajuste a relação entre qualidade da imagem e nível de compressão do arquivo.

O JPEG é atualmente o melhor método de compressão para imagens fotográficas, mas, por outro lado, o seu processo de compressão/descompressão é bastante lento. A compressão de uma imagem através do método de compressão JPEG se dá normalmente em cinco etapas fundamentais:

- 1 - Conversão da Imagem
- 2 – Sub-amostragem
- 3 - TDC – Transformada Discreta do Coseno
- 4 - Quantização
- 5 – Codificação da Entropia

A sigla BMP é a abreviatura da palavra em inglês *bitmap*, que significa mapa de bits. Este tipo de arquivo é uma representação fiel das informações contidas na imagem sem nenhum tipo de compressão.

A organização deste tipo de arquivo segue um padrão bem definido, sendo iniciado por um cabeçalho onde se encontram todas as informações necessárias para sua manipulação. Dependendo do número de bits da imagem pode existir uma tabela com os valores das cores (paleta de cores) antes do início dos dados.

2.5 ELEMENTOS DE UM SISTEMA DE PROCESSAMENTO DIGITAL DE IMAGENS

Segundo Gonzalez [6], um sistema de processamento digital de imagens é composto basicamente de cinco elementos:

(1) Aquisição de imagem:

Como os computadores podem processar apenas imagens codificadas em informações digitais, e as imagens na natureza encontram-se em outras formas, um pré-requisito para o processamento digital de imagens é a conversão de uma imagem para a forma digital [5]; a essa conversão dá-se o nome de aquisição ou digitalização de imagens.

O equipamento especializado usado para a aquisição é o que transforma um simples computador em uma estação de trabalho para processamento de imagens. Anteriormente, estes equipamentos eram caros, porém, com o avanço da tecnologia, estes dispositivos ficaram mais baratos e tornaram-se acessíveis para muitas aplicações.

Um digitalizador de imagens deve ter a capacidade de dividir uma imagem em *pixels* e endereçá-los 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 que isso seja possível um digitalizador precisa ter cinco elementos [5].

O primeiro elemento de um digitalizador é o sistema de amostragem. Este permite ao dispositivo acessar os *pixels* individualmente enquanto ignora o restante da imagem.

O segundo elemento é um mecanismo de varredura da imagem. Este processo consiste em mover o sistema de amostragem sobre toda a imagem dentro de um determinado padrão. A varredura permite ao sistema de amostragem endereçar os *pixels* em ordem, um por vez.

O terceiro elemento é o sensor de luz, o qual permite fazer a medida da intensidade do brilho da imagem em cada *pixel* através do sistema de amostragem. O sensor é normalmente um transdutor que converte intensidade luminosa em corrente ou tensão elétrica.

O quarto elemento, um quantificador, converte a saída contínua do sensor de luz em um valor inteiro. Tipicamente, o quantificador é um circuito eletrônico chamado de conversor analógico - digital. Esta unidade produz um número inteiro proporcional a tensão ou corrente de entrada.

O quinto e último elemento de um digitalizador é um armazenador. Os valores dos níveis de cinza produzidos pelo quantificador devem ser armazenados apropriadamente para processamento futuro.

(2) Armazenamento:

Uma imagem de 8 bits por *pixel* com dimensões de 1024x1024 *pixels* necessita de um milhão de bytes para seu armazenamento. Assim, fornecer um armazenamento adequado dos dados é uma tarefa desafiadora no projeto de sistemas de processamento de imagem. O armazenamento é medido em bytes, Kbytes (milhares de bytes), Mbytes (milhões de bytes), Gbytes (bilhões de bytes) e até Tbytes (trilhões de bytes).

(3) Processamento:

O processamento digital de imagens envolve procedimentos que são usualmente expressados em forma de algoritmos. Assim, com exceção da aquisição e apresentação da imagem, a maioria das funções de processamento de imagem pode ser implementada em *software* [6]. A única razão para *hardwares* específicos de processamento de imagem é a necessidade de um processamento mais veloz ou superação de alguma limitação computacional da máquina utilizada.

(4) Comunicação:

A comunicação em um sistema de processamento digital de imagens envolve comunicação local, entre o sistema de processamento de imagem, e comunicação remota de um ponto ao outro, tipicamente em conexões com a transmissão de dados das imagens.

(5) Apresentação:

A apresentação da imagem é o elemento final da cadeia de um sistema de processamento de imagem. Monitores monocromáticos e coloridos são os principais dispositivos de apresentação usados, sendo responsáveis por transformar a imagem em uma forma apropriada para a percepção humana. Além disso, impressoras coloridas e em tons de cinza, canhão de projeção, e outros dispositivos, também são utilizados.

A figura 2.5 mostra os elementos básicos de um sistema de processamento de imagens com exemplos de cada um dos tipos citados.

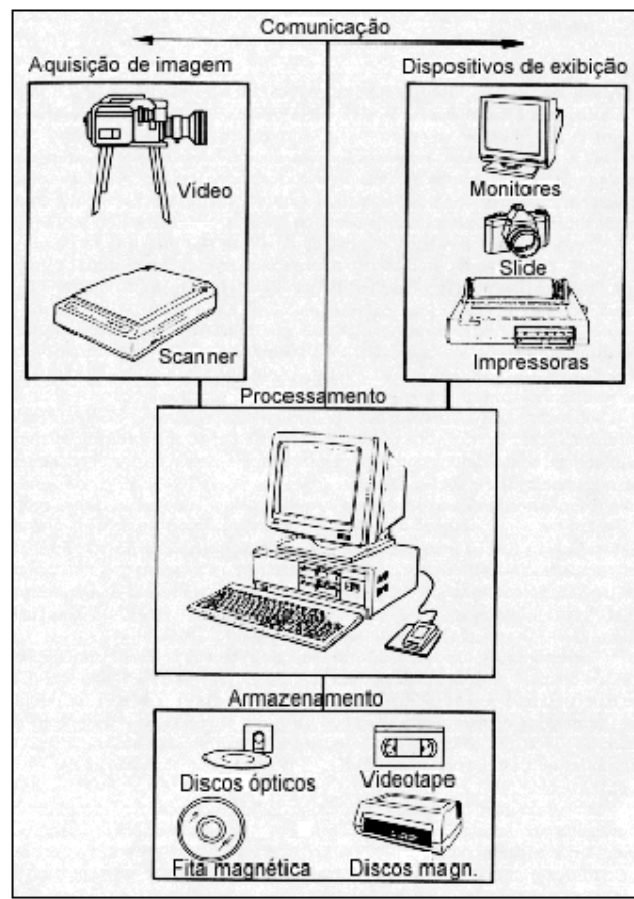


Fig. 2.5 – Elementos de um sistema de processamento e análise de imagem (adaptado Gonzalez [6] pg.11)

2.6 PROCESSAMENTO E ANÁLISE DE IMAGENS

Existem duas atividades no tratamento de imagem: o processamento e a análise. Segundo Castleman [5], o processamento inicia com uma imagem e produz como resultado uma versão modificada da mesma; este produto poderá ser submetido a outros processamentos ou uma análise, quer seja por parte de um usuário ou por parte de um sistema artificial. Exemplos da utilização do processamento de imagem são: correção da forma da imagem fora de foco, correção de cor, brilho ou contraste, suavização de ruído de fundo e, principalmente, o realce de estruturas que representem informações importantes.

A análise é feita sobre uma imagem normalmente resultante de um processamento. Porém, o resultado gerado é um conjunto de medidas ou até mesmo uma decisão. A análise de imagem geralmente extrai o objeto de interesse do restante da imagem com o intuito de caracterizá-lo, seja através de sua forma geométrica, ou através de suas características como cor, textura, etc.

O processamento de imagem, de um modo geral, pode ser classificado em três áreas:

(1) *Realce*: O principal objetivo das técnicas de realce é processar uma imagem qualquer de forma a obter um resultado mais apropriado do que a imagem original para uma aplicação específica [6]. Como exemplo de técnicas utilizadas nesta área tem-se: equalização de histograma, filtragem espacial e em frequência.

(2) *Restauração*: Como no realce de imagens, o objetivo final das técnicas de restauração é melhorar a qualidade de uma imagem em algum sentido [6]. Com o propósito de diferenciar entre realce e restauração considera-se esta como sendo um processo que tenta reconstruir ou recuperar uma imagem que tenha sido degradada, a priori, por algum fenômeno conhecido. Pertencem a esta área técnicas como: filtragem inversa e filtro de Wiener.

(3) *Compressão*: O objetivo das técnicas de compressão é a redução da quantidade de dados necessários para representação de uma determinada imagem, com ou sem perda de informação. A esta área pertencem técnicas como: codificação por comprimento variável e codificação por zona.

A segmentação, juntamente com a descrição, são consideradas áreas da análise de imagem.

(1) *Segmentação*: A segmentação subdivide uma imagem em suas partes constituintes ou objetos. Os algoritmos de segmentação, para imagens monocromáticas, geralmente são baseados em uma das duas propriedades básicas de imagens em tons de cinza: descontinuidades ou similaridades [6].

(2) *Descrição*: Após uma imagem ter sido segmentada em regiões, o conglomerado resultante dos *pixels* segmentados são, usualmente, representados e descritos de uma forma apropriada para posterior interpretação [6].

Após a imagem ser processada e analisada, normalmente uma interpretação dos resultados se faz necessária. As técnicas de inteligência artificial possibilitam o desenvolvimento de um sistema capaz de tomar decisões em função dos dados analisados. São algumas delas: redes neurais, algoritmos genéticos, lógica fuzzy, etc.

CAPÍTULO 3

PROCESSAMENTO DE IMAGEM

3.0 INTRODUÇÃO

No capítulo anterior foram vistos alguns conceitos e definições importantes. Neste ter-se-á como objetivo estudar técnicas de processamento de imagens utilizadas no desenvolvimento deste trabalho. Como está-se interessado na localização de caracteres alfanuméricos, estudar-se-ão apenas técnicas de realce, deixando de lado a restauração e compressão de imagens.

3.1 REALCE DE IMAGENS

O realce de uma imagem consiste na aplicação de um, ou uma seqüência de, algoritmos objetivando a melhora de sua aparência, seja para melhor visualização pelo sistema visual humano, ou para ressaltar determinadas características para uma posterior análise por um sistema de visão artificial. Quando as imagens são realçadas para visualização, como na televisão, o objetivo é a melhora de aspectos como: qualidade, inteligibilidade e aparência visual das mesmas. Em outras aplicações, como em sistemas de visão artificial, uma imagem é realçada para salientar características que serão analisadas em uma etapa posterior, ajudando na performance do sistema. Cabe salientar que todas as informações estão disponíveis na imagem original, portanto as técnicas de realce apenas evidenciam determinadas características de forma a torná-las mais facilmente utilizáveis.

O objetivo do realce é dependente da aplicação e o critério utilizado é subjetivo ou muito complexo. Isto faz com que as técnicas de realce sejam simples, porém específicas, ou seja, uma determinada técnica que funciona bem com uma classe de imagens pode não funcionar com outra classe.

3.1.0 Técnicas de realce por modificação no histograma

Uma das mais simples e úteis técnicas de realce é a modificação do histograma de uma imagem. O histograma resume o conteúdo, em tons de cinza, de uma imagem [5]. Enquanto em algumas imagens é possível retirar determinadas informações pela simples

análise de seu histograma, em outras é possível descrevê-la totalmente através do mesmo. A computação do histograma é uma tarefa simples e pode ser feita sem grandes custos computacionais.

3.1.1 Definição do histograma de uma imagem

O histograma é uma função que mostra, para cada tom de cinza, o número de *pixels* dentro da imagem que possui aquele tom. No eixo das abcissas tem-se os tons de cinza e no eixo das ordenadas a frequência de ocorrência. Na figura 3.1 tem-se uma imagem e seu histograma.

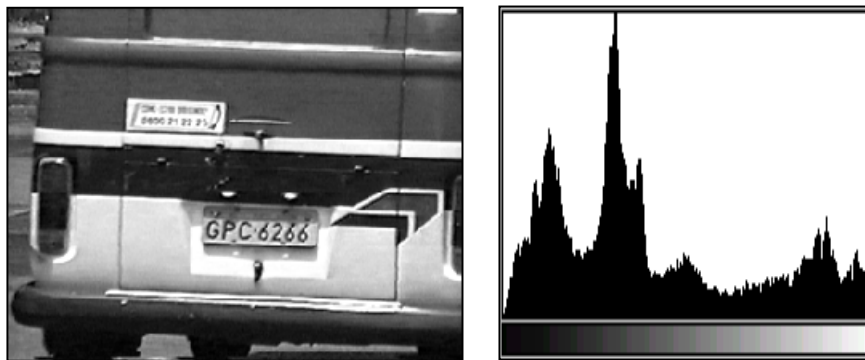


Fig. 3.1 – Imagem e histograma

Quando uma imagem é condensada em um histograma, toda a informação espacial é descartada [5]. O histograma especifica apenas a ocorrência de cada tom de cinza dentro da imagem, porém não diz onde eles estão localizados. Assim, o histograma é único para cada imagem, porém o reverso não é verdadeiro. Diferentes imagens podem possuir histogramas idênticos.

3.1.2 Alteração de contraste por modificação no histograma

Imagens com pouco contraste podem ser resultado de uma iluminação deficiente, ou algum outro tipo de falha no processo de aquisição. O objetivo desta técnica é aumentar ao máximo a relação de contraste entre os elementos que compõem a imagem, tornando mais evidente a separação dos mesmos. Para alterar o contraste de uma imagem é necessário expandir, ao máximo, a faixa de distribuição dos *pixels* utilizada na imagem original. Este tipo de técnica tende a preservar a ordem dos tons de cinza dentro da imagem, evitando o aparecimento de artefatos indevidos na imagem processada.

A imagem da figura 3.2 possui seu histograma “achatado” entre os níveis de cinza 0 (preto) e 60 (cinza escuro). Para melhor adequar esta imagem à visualização, uma alteração de contraste se faz necessária. Logo, uma modificação no seu histograma é feita com o intuito de expandir o histograma original de modo que ele ocupe toda faixa dos tons de cinza. Na figura 3.3 tem-se o resultado obtido por esta técnica, normalmente referenciada como *contrast stretching* [6].



Fig. 3.2 – Imagem com baixo contraste e seu histograma

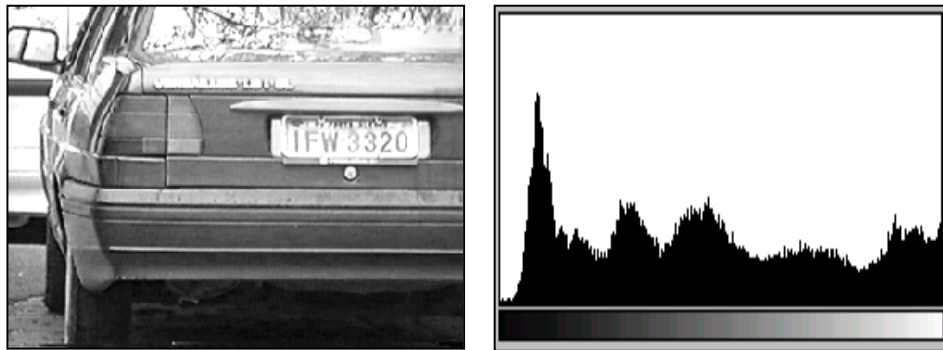


Fig. 3.3 – Imagem realçada e seu histograma

É importante salientar que na imagem original (Fig. 3.2) toda a informação está contida dentro da faixa mostrada no histograma, entre os tons de cinza 0 e 60. A utilização da técnica de realce, *contrast stretching*, apenas permitiu que esta informação se tornasse mais explícita, sem alterá-la, preservando a proporção de contraste entre os *pixels* da imagem.

Para entender-se como este procedimento funciona declara-se t como a função que leva a escala de tons de cinza i em uma escala de cinzas transformada $i' = t(i)$ com uma faixa de tons de cinza $[A,B]$. Se $f(x,y)$ é uma imagem com valores limitados em $[a,b] \subset [A,B]$, uma transformação linear que expande e desloca a faixa de tons de cinza para cobrir todo o espaço $[A,B]$, pode ser definida como:

$$i' = \frac{B-A}{b-a}(i-a) + A = \frac{B-A}{b-a}i + \frac{Ab-Ba}{b-a} \quad (3.1)$$

Uma análise mais prática da equação 3.1 fornece a seguinte função:

$$i' = \begin{cases} \frac{B-A}{b-a}i + \frac{Ab-Ba}{b-a}, & a \leq i \leq b \\ A, & i < a \\ B, & i > b \end{cases} \quad (3.2)$$

No caso da Fig 3.2, o intervalo $[a,b]$ é $[0,60]$ e o intervalo $[A,B]$ é $[0,255]$. Substituindo na Equação 3.1 tem-se:

$$i' = \frac{255-0}{60-0}(i-0) + 0 = \frac{255}{60}i = 4.25i$$

Ou seja, qualquer valor em i (Fig. 3.2) terá seu valor em i' (Fig.3.3) multiplicado por um fator de 4.25. Em função disso há um “esticamento” do histograma como pode ser observado. Algumas variações deste método são usualmente utilizadas. Em geral definem-se dois pontos de controle, (a,a') e (b,b') , conforme pode ser visto na Fig. 3.4. A condição para que não haja o aparecimento de artefatos indesejáveis [4] é que a curva permaneça monotonicamente crescente, em outras palavras, que $a < b$ e $a' < b'$.

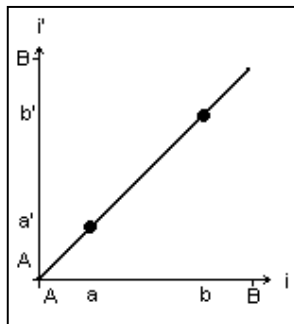


Fig. 3.4 – Variação do contrast stretching

A manipulação destes dois pontos permite diversas configurações para a alteração de contraste de uma imagem, sendo as mais comuns [9]:

- (a) $a > a'$ e $b < b'$
- (b) $a \leq a'$ e $b < b'$
- (c) $a > a'$ e $b \geq b'$
- (d) $a \leq a'$ e $b \geq b'$

Neste caso, tem-se:

$$i' = \begin{cases} f_1(i), & A \leq i < a \\ g(i), & a \leq i \leq b \\ f_2(i), & b < i \leq B \end{cases}$$

Com:

$$g(i) = \frac{B-A}{b-a}i + \frac{Ab-Ba}{b-a}$$

$$f_1(i) = \begin{cases} g(i) & \text{se } a \leq a' \\ \frac{a'}{a}i & \text{se } a > a' \end{cases}$$

$$f_2(i) = \begin{cases} g(i) & \text{se } b \geq b' \\ \frac{b'-B}{b-B}i + \frac{b-b'}{b-B}B & \text{se } b < b' \end{cases}$$

No algoritmo sugerido por Comelli [9] tem-se como constantes $a'=0.1 \times 255$, $b'=0.9 \times 255$, $A = 0$ e $B = 255$. Para sua implementação basta calcular-se os valores de a e b . Para isso, pode-se assumir que uma imagem possui $n \times m$ pixels, neste caso o valor a é o menor inteiro tal que:

$$\sum_{i=0}^a h(i) \geq \frac{nm}{10}$$

E o valor b é o maior inteiro tal que:

$$\sum_{i=0}^b h(i) \leq \frac{9nm}{10}$$

Este procedimento, aplicado de uma forma adequada, consegue aumentar a relação de contraste entre os caracteres e o fundo de uma placa de licença de um veículo.

CAPÍTULO 4

ANÁLISE DE IMAGEM

4.0 INTRODUÇÃO

Este capítulo se propõe a apresentar as técnicas de análise de imagem utilizadas no desenvolvimento do trabalho. A análise de uma imagem normalmente é feita após o processamento da mesma. Seu objetivo é separar os elementos de interesse, segundo algum critério, e descrevê-los através de suas características para permitir uma interpretação do seu conteúdo. Como mencionado anteriormente, ela pode ser dividida em duas etapas: segmentação e descrição.

4.1 SEGMENTAÇÃO DE IMAGENS

O primeiro passo na análise de imagens geralmente é a segmentação. Esta, por sua vez, subdivide uma imagem em suas partes constituintes. O nível de subdivisão depende do problema a ser resolvido, isto é, a etapa de segmentação só termina quando os objetos de interesse da aplicação em questão são isolados.

Em geral, a segmentação automática é uma das mais difíceis tarefas [6] em sistemas de processamento e análise de imagens digitais. Este passo é crucial e determina um eventual sucesso ou falha na análise final de uma imagem. De fato, uma segmentação efetiva raramente conduz a uma interpretação errada.

Os algoritmos de segmentação para imagens monocromáticas são geralmente baseados em discontinuidades ou similaridades entre *pixels*. Na análise por discontinuidades, a abordagem é particionar a imagem em função de mudanças abruptas de níveis de cinza. A principal área de interesse desta categoria é a detecção de pontos isolados e a detecção de linhas e bordas de uma imagem, sendo muito útil na determinação de contornos de objetos.

Na análise por similaridade, há uma procura por regiões da imagem que possuam um mesmo padrão, por exemplo cor ou textura. Isto é feito através da utilização de técnicas de limiarização, crescimento, divisão e fusão de regiões [6].

Imagens binárias – aquelas que possuem somente dois níveis de intensidade – constituem um importante subconjunto de imagens digitais, e normalmente resultam de uma operação de segmentação. A qualidade da etapa de binarização é crítica para análises subseqüentes [10].

A limiarização, ou *thresholding*, uma das mais importantes abordagens da segmentação de imagens [6], é uma técnica de análise por região particularmente útil para cenas que contém objetos sobre um fundo contrastante. Sua implementação computacional geralmente é simples, e ela nunca falha para separar regiões fechadas distintas [5].

Quando o valor do *pixel* analisado é maior, ou menor, que o valor definido como limiar, o *pixel* é considerado pertencente ao objeto, do contrário, se o *pixel* for menor, ou maior, que o valor de limiar, ele é considerado pertencente ao fundo. As técnicas de limiar podem ser divididas em duas categorias: global e local.

4.1.0 Binarização com limiar global:

A mais conhecida e simples técnica de binarização faz uso de um valor de limiar global. Na implementação desta técnica, o tom de cinza escolhido como limiar é mantido constante durante a análise de toda a imagem. Isso faz com que sua implementação seja fácil e seu custo computacional baixo.

Supondo que uma imagem contenha um objeto escuro e um fundo claro, a figura 4.1 ilustra o histograma de uma imagem com as características descritas.

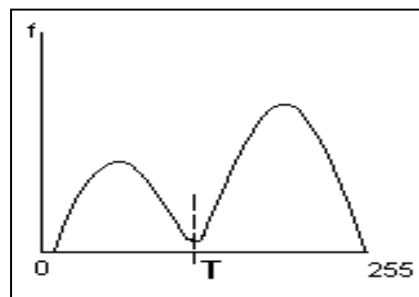


Fig. 4.1 – Histograma bimodal

A forma do histograma desta imagem é claramente bimodal, ou seja, possui dois modos distintos, um no pico escuro, onde tem-se o objeto, e o outro no pico claro, onde se encontra o fundo. A escolha de um valor de limiar sobre o vale do histograma

permite separar o fundo do objeto de interesse. Como exemplo real da utilização desta técnica veja-se a figura 4.2.

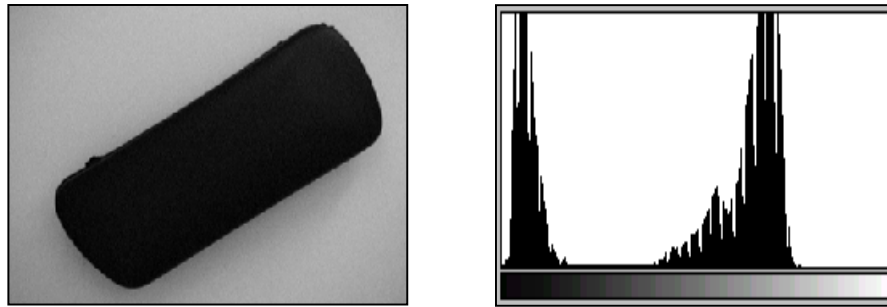


Fig. 4.2 – Imagem com histograma bimodal

Uma observação rápida sobre o histograma da imagem permite deduzir que trata-se de um caso de histograma bimodal. Portanto, para separar a imagem em apenas dois elementos: o objeto e o fundo, basta escolher um valor de limiar sobre o vale do histograma– região entre picos. Na figura 4.3 vê-se o resultado da aplicação da técnica de limiar global utilizando como limiar um valor igual a 60. Percebe-se que a forma, bem como o conteúdo do objeto de interesse, foram preservados.

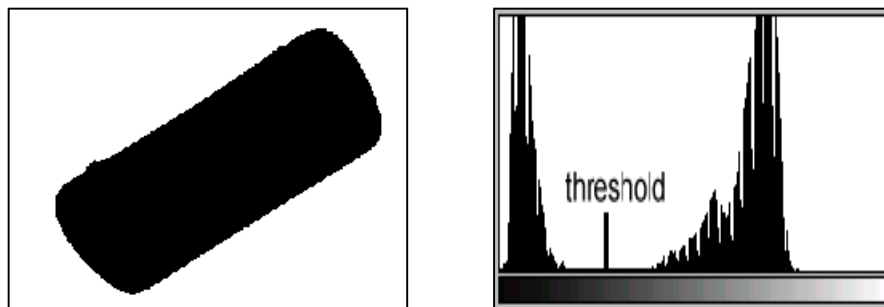


Fig. 4.3 – Resultado de uma binarização por *threshold* global

É evidente que a escolha adequada do valor de limiar é essencial para o bom funcionamento da técnica, e ainda, esta escolha é única para cada imagem. Isto explica os numerosos esforços feitos no desenvolvimento de algoritmos de limiarização automática.

4.1.1 Binarização local adaptativa:

Em muitos casos, principalmente quando não há um controle da iluminação sobre a imagem, o fundo não possui uma intensidade luminosa constante, e o contraste dos objetos varia com a imagem, como é o caso da figura 4.4.

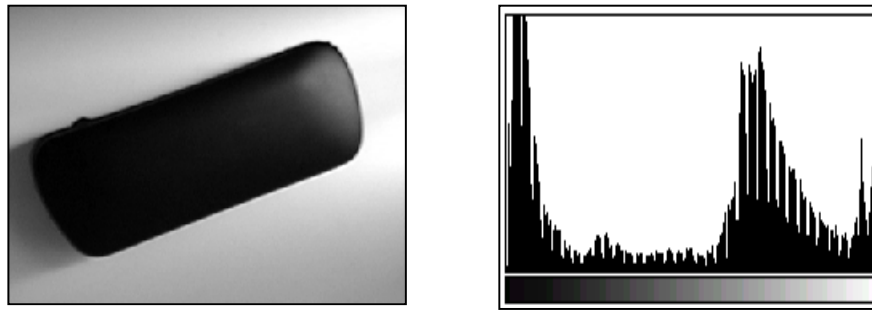


Fig. 4.4– Imagem com iluminação não uniforme e histograma

Nestes casos, um valor de limiar que fornece um bom resultado em uma determinada região pode não ser adequado em outra (Fig. 4.5), causando uma separação errada na imagem resultante.

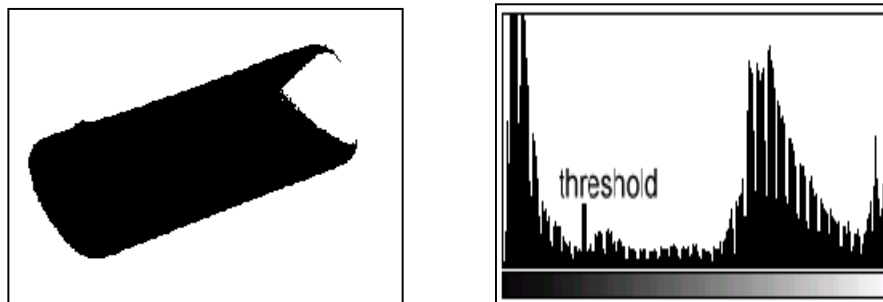


Fig. 4.5 – Resultado de uma binarização por *threshold* global

Como a iluminação sobre a superfície do objeto não é homogênea, um limiar global não funciona bem causando uma perda de informação na forma do objeto. É necessário um limiar variável que se adapte as diferentes condições de iluminação na imagem.

Uma das aplicações mais comuns deste tipo de binarização é no campo da análise de imagens de documentos [11]. A binarização de imagens em escala de cinza digitalizadas é o primeiro passo na maioria dos sistemas de análise de imagens de documentos, e a seleção de um método de binarização apropriado para esta tarefa é um problema não trivial.

Em um estudo feito por Trier [11], cujo objetivo era o reconhecimento de caracteres, a binarização *Niblack* [8] mostrou-se superior a outros onze métodos de binarização local adaptativa analisados.

4.1.2 Binarização *Niblack*

O algoritmo de binarização *Niblack*, um dos algoritmos de binarização local adaptativa existentes, é baseado em uma relação estatística entre cada *pixel* analisado e sua vizinhança.

O algoritmo calcula para cada *pixel* da imagem a média (4.1) e o desvio padrão (4.2) da vizinhança em torno dele, e a seguir, compara o valor do *pixel* analisado com o limiar $T(x,y)$ (4.3). Se o valor do *pixel* for maior que $T(x,y)$, ele é considerado pertencente ao fundo, caso contrário, é considerado pertencente a algum objeto.

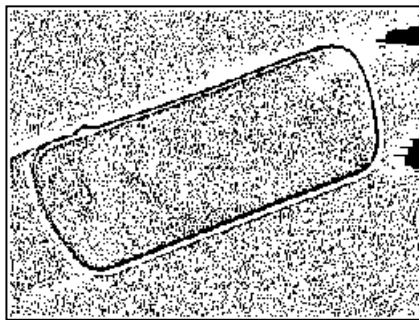
$$\mu(x, y) = \frac{1}{N.M} \sum_{i=0}^N \sum_{j=0}^M p(x, y) \quad (4.1)$$

$$\sigma(x, y) = \sqrt{\frac{1}{N.M} \sum_{i=0}^N \sum_{j=0}^M (p(x, y) - \mu(x, y))^2} \quad (4.2)$$

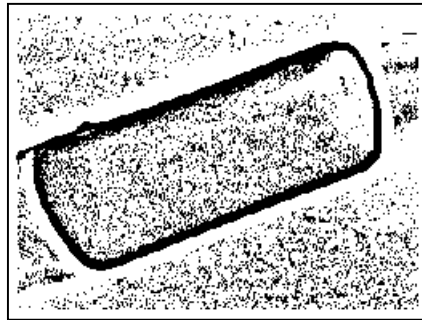
$$T(x, y) = -\alpha.\sigma(x, y) + \mu(x, y) \quad (4.3)$$

O tamanho da janela precisa ser pequeno o bastante para preservar detalhes locais, e suficientemente grande para suprimir ruídos indesejáveis. Nos testes feitos por Trier [11], bem como na literatura pesquisada, é padrão a utilização de uma janela quadrada de dimensões 15×15 *pixels* e uma valor de α constante igual a 0.2.

Na figura 4.6 vê-se o resultado da binarização *Niblack*, para a mesma imagem apresentada na figura 4.4, utilizando o valor de α fixo em 0.2 e uma janela de amostragem quadrada com dimensões de 5, 15, 30 e 60 *pixels* respectivamente.



Janela 5x5 *pixels*



Janela 15x15 *pixels*

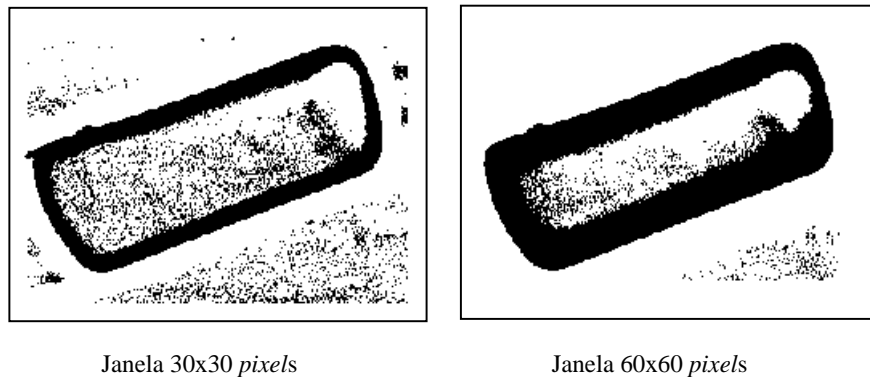


Fig. 4.6 – Resultado da binarização *Niblack* para diferentes janelas de amostragem

Duas conclusões importantes podem ser extraídas da análise das imagens acima. A primeira é que, à medida que a janela de amostragem cresce, o ruído vai sendo suprimido. Isto ocorre porque a binarização *Niblack* opera em cima da distribuição estatística dos *pixels* e suas vizinhanças, portanto, a amostra analisada deve possuir dimensões próximas das dimensões do objeto que se procura segmentar. A segunda conclusão é que as formas originais do objeto permanecem inalteradas para qualquer uma das dimensões utilizadas. Isto ocorre porque nas bordas do objeto a janela de amostragem tem, em média, metade de seus *pixels* preenchidos pelos *pixels* do objeto e a outra metade pelos *pixels* do fundo.

4.2 REPRESENTAÇÃO E DESCRIÇÃO DE OBJETOS

Após uma imagem ter sido segmentada em regiões, quer seja pela análise de descontinuidades, quer seja pela análise de similaridade, o conglomerado de *pixels* resultantes são normalmente representados e descritos para um posterior processamento. Basicamente, representar uma região envolve duas escolhas: representá-la através de suas características externas (como bordas), ou representá-la em termos de suas características internas (os *pixels* englobados pela região).

Geralmente, uma representação através de características externas é utilizada quando o interesse está na forma do objeto. Em contrapartida, quando o foco de interesse está sobre propriedades reflectivas, como cor e textura, uma representação em termos das características internas é usada. Há casos em que se está mais interessado em um determinado tipo de descrição devido a suas propriedades, por exemplo, uma descrição

que possua ao mesmo tempo invariância ao tamanho, rotação e translação de objetos, como é o caso da descrição através dos coeficientes de Fourier.

4.2.0 Representação por projeção vertical e horizontal

Um tipo de representação proposto neste trabalho, é a representação por projeção. A idéia básica por trás desta representação, que faz uso das características externas do objeto de interesse, é caracterizar um determinado objeto, no caso caracteres, através de suas características espaciais utilizando suas projeções verticais e horizontais. Esta representação foi criada para auxiliar a rede neural encarregada do reconhecimento dos caracteres e foram utilizadas amostras dos caracteres conjuntamente com sua descrição (Fig. 4.7).

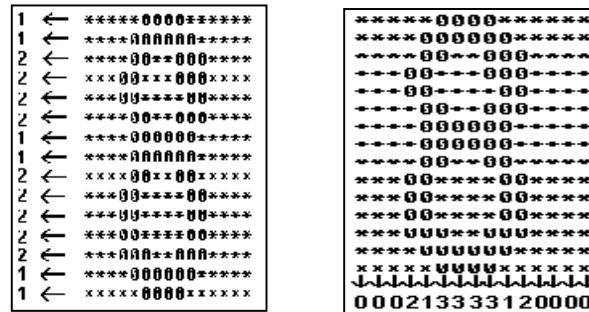


Fig. 4.7 – Projeções horizontais e verticais

Na figura 4.7 tem-se à esquerda a projeção vertical do caracter 8. Cada número representa a quantidade de segmentos de zeros encontrados na respectiva linha. O mesmo acontece na segunda figura, onde se tem a projeção horizontal. O par de linhas gerado pela descrição segue um padrão único para cada caracter.

CAPÍTULO 5

RECONHECIMENTO E INTERPRETAÇÃO DE IMAGEM

5.0 INTRODUÇÃO

Um dos objetivos de um sistema automático de análise de imagem por computador é dotar uma máquina de alguma capacidade de visão, de modo que o sistema possa desempenhar tarefas de uma forma semelhante ao ser humano para aquela aplicação. Por exemplo, em um sistema para leitura automática de imagens de documentos, o padrão de interesse são caracteres alfanuméricos, e o objetivo do sistema é conseguir uma precisão no reconhecimento dos caracteres o mais próxima possível da soberba capacidade demonstrada pelos seres humanos para executar este tipo de tarefa [6].

Um sistema automático de análise de imagem deve ser capaz de exibir vários graus de inteligência. O conceito de inteligência é algo um tanto vago, particularmente quando se refere a máquinas. Entretanto, mencionar diversos tipos de comportamento geralmente associados com inteligência não é difícil. São alguns deles: (1) a habilidade de extrair informações pertinentes de um contexto com detalhes irrelevantes; (2) a capacidade de aprender através de exemplos e generalizar este conhecimento para aplicar em diferentes circunstâncias; (3) a habilidade de fazer inferências em informações incompletas.

Sistemas de análise de imagem com estas características podem ser projetados e implementados para ambientes operacionais limitados, entretanto, ainda não se sabe como dotar estes sistemas com um desempenho que esteja próximo de simular a capacidade humana em aplicações genéricas. As contínuas pesquisas em biologia e sistemas computacionais tem descoberto novas e promissoras teorias para explicar o conhecimento visual humano. Porém, o estado da arte em análise de imagem por computador, em geral, tem-se baseado em formulações heurísticas aplicáveis em problemas específicos [6].

Por exemplo, algumas máquinas são capazes de ler documentos impressos, apropriadamente formatados, muito mais rápido que qualquer ser humano. No entanto,

sistemas deste tipo são altamente especializados e possuem pouca, ou nenhuma capacidade de extensão.

5.1 ELEMENTOS DE ANÁLISE DE IMAGEM

Segundo Gonzalez [6], a análise de imagem pode ser dividida em três áreas. São elas: (1) processamento de baixo-nível, (2) processamento de nível intermediário e (3) processamento de alto nível. Embora não existam limites bem definidos entre elas, esta divisão nos fornece um esqueleto útil para distinguir os vários processos que são componentes inerentes de um sistema de análise de imagem autônomo. A figura 5.1 ilustra esta divisão.

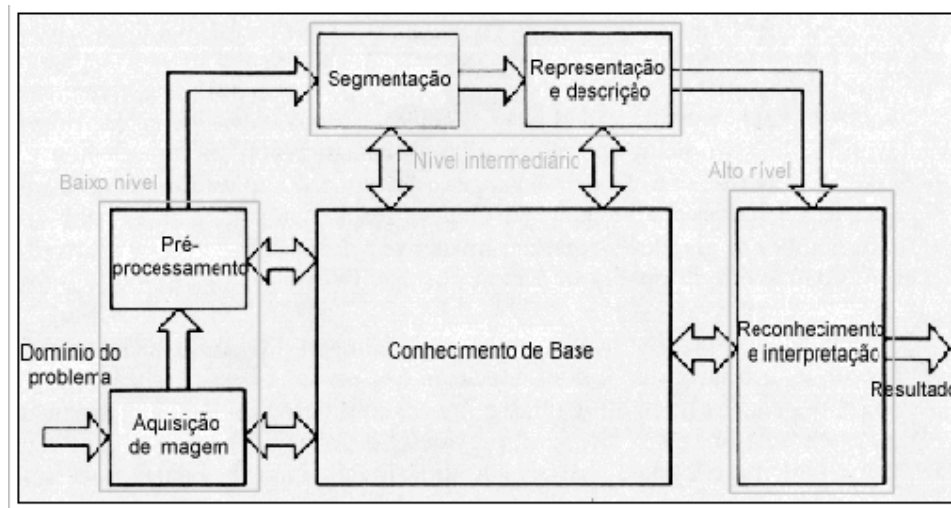


Fig. 5.1 – Elementos de análise de imagem (adaptado Gonzalez [6] pg. 572)

Processamento de baixo nível: É dividido em funções que podem ser vistas como reações automáticas, sem necessitar de inteligência. Os processos de aquisição e pré-processamento de imagem são considerados funções de baixo nível. Esta classificação engloba o processo de formação da imagem até a compensação da mesma através da redução de ruídos existentes ou melhora do aspecto visual. O processamento de baixo nível pode ser comparado aos processos de percepção e adaptação que um ser humano passa quando entra em um teatro escuro vindo de um dia ensolarado. O processo “inteligente” de encontrar um assento vago, não pode ser efetuado sem que haja uma formação apropriada da imagem. A ação desempenhada pelo cérebro humano na adaptação do sistema de visão para produzir esta imagem é automática; uma reação inconsciente.

Processamento de nível intermediário: É dividido em tarefas de extração e caracterização de componentes em uma imagem resultante de um processamento de baixo nível. A segmentação, representação e descrição de elementos pertencem a este nível.

Processamento de alto nível: Este processamento envolve o reconhecimento e a interpretação dos resultados conseguidos na etapa anterior. A maioria das técnicas utilizadas no processamento de nível baixo e intermediário, englobam um lógico e bem definido conjunto de formulações teóricas. Porém, no reconhecimento, e principalmente na interpretação, nosso conhecimento e entendimento de princípios fundamentais se tornam imprecisos e muito mais especulativos [6].

5.2 PROCESSAMENTO DE ALTO NÍVEL

No processamento de alto nível, existem três principais abordagens utilizadas para reconhecimento e interpretação de imagens. A primeira, chamada “métodos de decisão especulativos”, é baseada na representação de padrões sob a forma de vetores. Este método busca aproximar e agrupar vetores semelhantes, criando diferentes classes de padrões. Pertencem a esta abordagem métodos como classificação por mínima distância, correlação, classificador de Bayes e redes neurais. Na segunda abordagem, reconhecimento estrutural, os padrões são representados através de uma forma simbólica, e o reconhecimento é baseado em comparação ou em modelos que tratam os símbolos como sentenças de uma linguagem artificial. A terceira e última abordagem, trabalha com a atribuição de significados a um grupo de elementos de imagem reconhecidos [6].

A interpretação de imagem é uma das aplicações mais significativas de redes neurais [12]. Diversos sistemas estudados fizeram uso desta técnica na etapa de reconhecimento dos caracteres. O principal objetivo deste trabalho é o desenvolvimento de um sistema de visão artificial capaz de identificar automaticamente veículos, através da localização e leitura de sua placa. Foi considerada como prioritária a tarefa de localização e segmentação dos caracteres, tarefa que exige a utilização de técnicas de processamento, e análise de imagens. Por este motivo, a tarefa de leitura não foi explorada com a mesma intensidade que as outras, e foi escolhido o método de reconhecimento por redes neurais.

As vantagens mais freqüentemente citadas, em favor da abordagem por redes neurais na solução de problemas de reconhecimento de padrões são que: (1) ela requer menos conhecimento do problema que outras abordagens, (2) ela é capaz de implementar um particionamento mais complexo do espaço de características, e (3) ela possui alto desempenho em implementações de processamento paralelo. Defensores dessa abordagem também ressaltam que por se assemelhar ao modo como funciona o cérebro humano, as redes neurais teriam potencial para aproximar-se de seu desempenho. Entretanto, testes mostram que a atuação da rede, nestes casos, tende a se aproximar de classificadores estatísticos bem treinados [5].

5.3 Redes Neurais

A diferença entre o modo como o cérebro humano e o computador trabalham motivaram o estudo das rede neurais artificiais. Tipicamente, neurônios são cinco a seis ordens de grandeza mais lentos que as portas lógicas de silício; eventos em um chip de silício ocorrem em nanosegundos ($10E-9s$), enquanto que em neurônios ocorrem em milisegundos ($10E-3s$). Entretanto, o enorme número de neurônios e suas interconexões (estima-se que sejam na ordem de 10 bilhões de neurônios no córtex humano e 60 trilhões de sinapses) faz do cérebro humano uma estrutura extremamente eficiente. Especialmente quanto à eficiência energética, que é de aproximadamente $10E-16$ joules por operação por segundo, contra, para os melhores computadores, algo em torno de $10E-6$ joules por operação por segundo [13].

O cérebro humano é complexo, não linear e paralelo em seu processamento. Ele possui a capacidade de organizar seus neurônios para desempenhar determinadas tarefas (i.e. reconhecimento de padrões, percepção e controle motor) muitas vezes mais rápido que o mais rápido computador existente hoje. O interesse pelo funcionamento do cérebro fez com que os pesquisadores criassem modelos que se assemelhassem ao comportamento dos neurônios.

A visão por computador tem sido, há muito tempo, um importante campo de pesquisa em inteligência artificial. Esta pesquisa é motivada pela necessidade de sistemas flexíveis capazes de desempenhar tarefas complexas como reconhecimento, inspeção e navegação em diversas aplicações que vão desde reconhecimento óptico de

caracteres, em automação postal a sistemas de produção integrados por computadores em plantas industriais [14].

As redes neurais têm sido vistas como uma ferramenta útil para diferentes tipos de aplicações da visão por computador. Muito da pesquisa atual em redes neurais é centrada em modelos de redes individuais, enquanto que em complicadas tarefas de engenharia, como a visão por computador, um sistema misto é mais desejado. As redes individuais são então vistas como componentes de um sistema maior que contém ainda tradicionais técnicas de processamento de imagens, como filtragens, segmentação, etc. Este tipo de uso conduz a uma arquitetura híbrida na qual alguns dos módulos de processamento são baseados em redes neurais. Para este tipo de sistema o problema é decidir quais os benefícios que a utilização de redes neurais pode trazer para a visão por computador, qual o nível mais apropriado para sua utilização e que tipo de modelo de rede deve ser usado.

Uma rede neural é um processador distribuído massivamente paralelo que possui uma propensão natural para armazenar conhecimento adquirido por experiência e disponibilizá-lo para uso [13]. Ela se assemelha ao cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede através de um processo de aprendizado.
2. As forças de conexão entre neurônios, conhecidas como pesos sinápticos, são usadas para o armazenamento do conhecimento.

5.3.1 Modelo de um neurônio

O neurônio é a unidade de processamento de informações fundamental para a operação de uma rede neural. No seu modelo (Fig. 5.2) podem-se identificar três elementos básicos:

1. Um conjunto de sinapses ou conexões, cada uma caracterizada por um peso w_{kp} . O peso é positivo se a sinapse é excitatória e negativo se é inibitória.
2. Um somador que têm como resultado u_k a soma dos sinais de entrada x_p multiplicados pelos pesos w_{kp} das respectivas sinapses.

3. Uma função de ativação para determinação da saída $\varphi(\cdot)$. Tipicamente, a faixa normalizada de amplitude de saída desta função fica entre o intervalo $[0,1]$, ou, alternativamente entre $[-1,1]$.

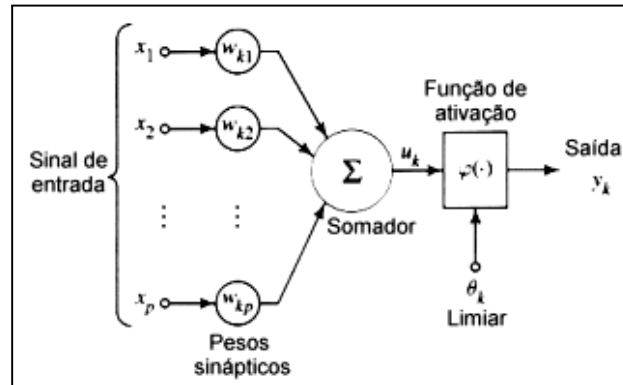


Fig. 5.2 – Modelo não linear de um neurônio (adaptado Haykin [13] pg. 8)

O modelo de um neurônio também inclui um limiar θ_k externamente aplicado que funciona como um nível DC positivo na função de ativação, ou nível DC negativo, sendo então denominado polarização. Matematicamente, podemos descrever um neurônio k pelo seguinte par de equações (Eq. 5.1):

$$u_k = \sum_{j=1}^p w_{kj} x_j \tag{5.1}$$

$$y_k = \varphi(u_k - \theta_k)$$

O elemento de processamento básico de uma rede neural, o neurônio, opera de uma forma simples. Ele soma o produto do vetor de entrada (x_p) pelo vetor de pesos (w_{kp}), transforma o resultado de acordo com uma função sigmoideal $\varphi(\cdot)$, e apresenta em sua saída (y_k) como resultado um valor escalar.

5.3.2 Arquitetura da rede

Uma rede neural é um conjunto de neurônios idênticos interconectados. O modo como eles estão conectados, ou a arquitetura da rede, é uma das maiores decisões de projeto e normalmente varia de aplicação para aplicação. Tipicamente, os neurônios são divididos em diferentes camadas (*layers*). A escolha do número de camadas, e neurônios contidos nela, também é uma decisão de projeto. Em algumas arquiteturas de rede, os neurônios contidos em uma camada só comunicam-se com os neurônios

contidos na camada posterior. Entretanto em outras, há uma comunicação entre os neurônios de uma mesma camada, ou até mesmo, uma realimentação entre camadas.

A primeira camada é chamada de camada de entrada, a última camada é conhecida como camada de saída, e as camadas intermediárias são denominadas camadas escondidas. A figura 5.3 ilustra algumas topologias conhecidas.

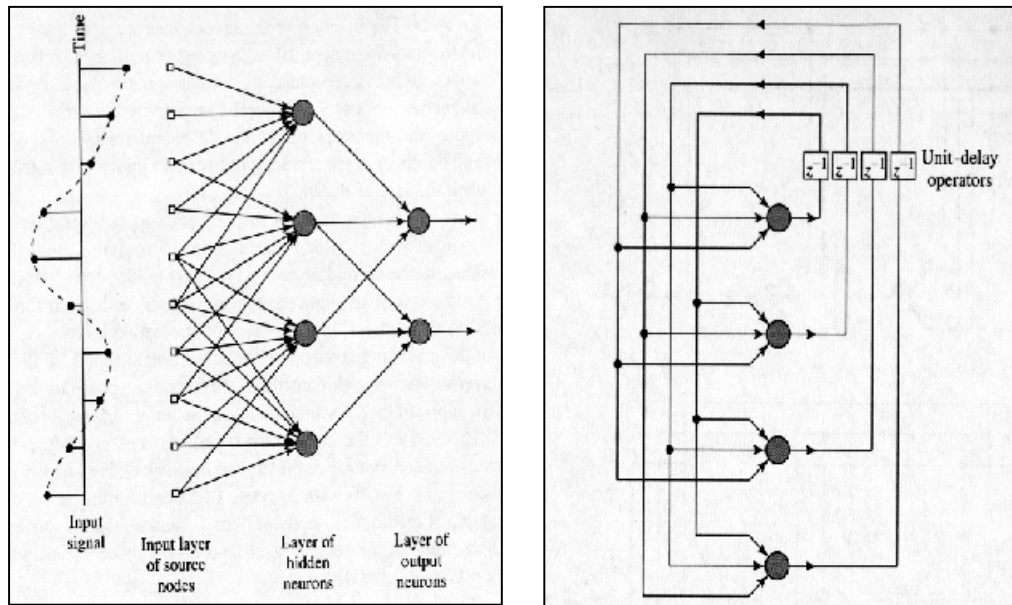


Fig. 5.3 – Topologia *feedforward* e recorrente (fonte Haykin [13] pg. 21)

5.3.3 Processo de aprendizado

Dentre as muitas propriedades interessantes de uma rede neural, a propriedade de maior importância é a habilidade da rede em aprender em seu ambiente, e melhorar seu desempenho em função disso. Uma rede neural aprende através de um processo iterativo de ajustes aplicados aos pesos das sinapses e limiares. Define-se o sentido da palavra aprender dentro do contexto de redes neurais na forma a seguir.

Aprender é o processo pelo qual parâmetros livres de uma rede neural são adaptados, através de um contínuo processo de estimulação pelo ambiente no qual a rede está imersa. O tipo de aprendizado é determinado pelo modo que os parâmetros são alterados [13].

Esta definição do processo de aprendizado implica na seguinte seqüência de eventos:

1. A rede neural é estimulada pelo ambiente.
2. A rede neural passa por mudanças em seus pesos sinápticos como resultado de estímulos.
3. A rede neural responde de uma nova maneira ao ambiente, por causa das mudanças que ocorreram em sua estrutura interna.

Durante um exercício de treinamento supervisionado, vetores de características de objetos conhecidos são apresentados inúmeras vezes, em ordem aleatória, à rede. Os pesos das sinapses são levemente alterados a cada iteração usando uma regra fixa de treinamento para conduzir a saída da rede em direção a valores corretos.

A velocidade de aprendizado é fortemente dependente da forma da superfície formada pela medida do erro no “espaço de pesos”. Este espaço tem uma dimensão para cada peso da rede e uma dimensão adicional que representa o erro total no desempenho da rede para um dado conjunto de pesos [15]. Para muitas tarefas, a superfície de erro possui ravinas que causam problemas para procedimentos que se baseiam na procura do erro mínimo simplesmente pela medida do gradiente descendente.

5.3.4 Operação da rede

Em aplicações de reconhecimento de padrões, a entrada da rede é um vetor de características de um objeto desconhecido. O vetor é apresentado a cada neurônio da primeira camada da rede e a informação é propagada através das várias camadas até que um vetor de saída apareça na camada de saída. O objeto desconhecido é atribuído a uma classe determinada pelo vetor de saída. A rede aceita um vetor de características como entrada e gera um vetor de saída indicando um valor correspondente à classe que o objeto pertence.

5.3.5 Redes neurais multilayer *feedforward*

O reconhecimento de caracteres é uma aplicação que pode ser resolvida com diferentes arquiteturas de rede, porém a topologia *feedforward* com algoritmo de treinamento *back propagation* fornece uma alternativa simples e eficiente. Esta topologia é utilizada quando existe um número de classes fixo e conhecido, o que é o caso dos caracteres alfanuméricos utilizados nas placas dos veículos.

Uma rede multilayer *feedforward* possui três características distintas:

1) O modelo de cada neurônio da rede inclui uma não-linearidade na sua saída. É importante enfatizar que esta não-linearidade é suave, diferenciável em qualquer ponto. Uma forma de não-linearidade normalmente utilizada que satisfaz a condição acima é uma não-linearidade sigmoideal definida pela função tangente hiperbólica.

2) A rede contém uma ou mais camadas escondidas (*hidden layers*). Estas camadas habilitam a rede a aprender tarefas complexas pela extração progressiva das principais características do vetor de entrada.

3) A rede exibe um alto grau de conectividade, determinado pelas sinapses. Uma mudança na conectividade da rede requer uma mudança na população de conexões sinápticas ou de seus pesos. De fato, é através da combinação destas características com a habilidade de aprender de experiências através de um treinamento que estas redes derivam sua força computacional.

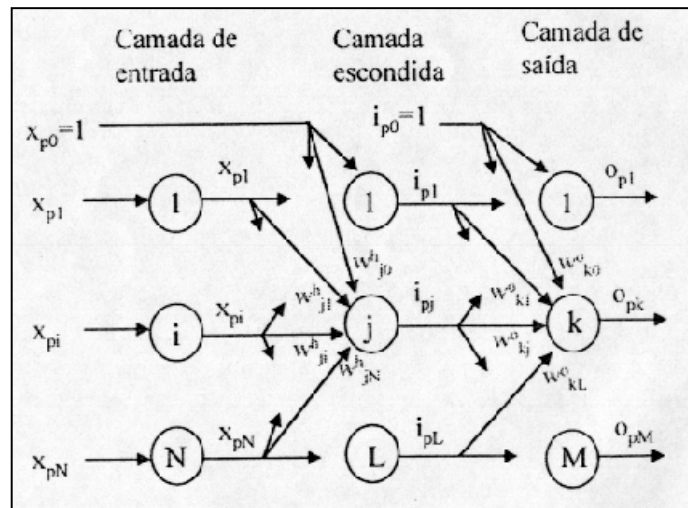


Fig. 5.4 – Topologia da rede *feedforward*

Tipicamente, a rede consiste de um conjunto de neurônios que constituem a camada de entrada (X_p), uma ou mais camadas escondidas (I_p) e uma camada de saída (O_p). O sinal de entrada se propaga através da rede em direção à saída da mesma, camada por camada. As redes neurais *multilayer feedforward* têm sido aplicadas com sucesso na resolução de diversos problemas. Este tipo de rede utiliza um algoritmo de treinamento supervisionado denominado *error back propagation*. Este algoritmo é baseado na regra

delta de aprendizado por correção de erro que pode ser vista como uma generalização de um popular algoritmo de filtragem adaptativa: (LMS) *least-mean-square* [13].

Se existe um finito e fixo conjunto de par de vetores (entrada-saída), o erro total do desempenho da rede, com um particular conjunto de pesos, pode ser calculado pela comparação dos vetores de saída atual e desejada para cada caso. Como existem vários neurônios na camada de saída, deve-se definir a soma instantânea dos quadrados dos erros em cada nó de saída da rede, E_p (5.2), quando o p -ésimo vetor de treinamento x_p é apresentado na entrada da mesma:

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (5.2)$$

Define-se o erro quadrático instantâneo na unidade k de saída como:

$$\delta_{pk}^2 = (y_{pk} - o_{pk})^2 \quad (5.3)$$

sendo δ_{pk} o erro em uma unidade de saída k quando um vetor x_p é apresentado na entrada, y_{pk} a saída desejada e o_{pk} a saída instantânea:

$$\delta_{pk} = (y_{pk} - o_{pk}) \quad (5.4)$$

O erro quadrado instantâneo (E_p) na saída da rede é usado no algoritmo *back propagation* como aproximação do erro médio quadrático (E_{av}). Este, por sua vez, é obtido da média dos erros instantâneos para todos os P vetores de treinamento:

$$E_{av} = \frac{1}{P} \sum_{p=1}^P E_p$$

A modificação dos pesos é feita a cada apresentação de um vetor de entrada. Cada peso é ajustado de acordo com o seu respectivo erro (5.5), de forma a produzir deslocamentos na direção do gradiente descendente do erro E_p . Observando o princípio da mínima perturbação, o erro médio quadrático será minimizado com este procedimento.

$$W_{k+1} = W_k + \eta(-\nabla E_p) \quad (5.5)$$

Na camada de saída, pode-se escrever o erro a ser minimizado como:

$$E_p = \frac{1}{2} \sum_{k=1}^M \sum_{p=1}^K (y_{pk} - o_{pk})^2$$

$$= \frac{1}{2} [(y_{p1} - o_{p1})^2 + \dots + (y_{pk} - o_{pk})^2 + \dots + (y_{pM} - o_{pM})^2]$$

Com $o_{pk} = f(\text{net}_{pk}^o)$,

sendo $\text{net}_{pk}^o = \sum_{j=0}^L w_{kj} \cdot i_{pj}$ $w_{k0} \cdot i_{p0} + \dots + w_{kj} \cdot i_{pj} + \dots + w_{kL} \cdot i_{pL}$.

A componente do gradiente em relação a um peso específico w_{kj}^o da camada de saída pode ser determinada pela regra da cadeia das derivadas parciais:

$$\frac{\partial E_p}{\partial w_{kj}^o} = \frac{\partial E_p}{\partial \delta_{pk}} \frac{\partial \delta_{pk}}{\partial o_{pk}} \frac{\partial o_{pk}}{\partial \text{net}_{pk}^o} \frac{\partial \text{net}_{pk}^o}{\partial w_{kj}^o}$$

Das definições anteriores, pode-se determinar as derivadas parciais:

$$\frac{\partial E_p}{\partial \delta_{pk}} = \delta_{pk}, \quad \frac{\partial \delta_{pk}}{\partial o_{pk}} = -1, \quad \frac{\partial o_{pk}}{\partial \text{net}_{pk}^o} = f'(\text{net}_{pk}^o) \quad \text{e} \quad \frac{\partial \text{net}_{pk}^o}{\partial w_{kj}^o} = i_{pj}$$

Com isso, a componente do gradiente do erro em relação a um peso específico da camada de saída w_{kj}^o pode ser determinada:

$$\frac{\partial E_p}{\partial w_{kj}^o} = -(y_{pk} - o_{pk}) \cdot f'(\text{net}_{pk}^o) \cdot i_{pj} = \delta_{pk} \cdot i_{pj}$$

Definindo-se a parcela de erro do neurônio de saída k como:

$$\delta_{pk} = (y_{pk} - o_{pk}) \cdot f'(\text{net}_{pk}^o), \quad (5.6)$$

logo, a expressão para atualização de um peso genérico na camada de saída fica:

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta(-\nabla E_p) = w_{kj}^o(t) + \eta \delta_{pk} \cdot i_{pj}, \quad (5.7)$$

e para a tangente hiperbólica como função de ativação tem-se:

$$\delta_{pk} = (y_{pk} - o_{pk}) \cdot (1 - o_{pk}^2) \quad (5.8)$$

Um procedimento análogo, porém mais trabalhoso, é feito para se encontrar a expressão de atualização de um peso genérico na camada escondida:

$$w^h_{ji}(t+1) = w^h_{ji}(t) + \eta(-\nabla E_p) = w^h_{ji}(t) + \eta \delta^h_{pj} \cdot x_{pi}, \quad (5.9)$$

com

$$\delta^h_{pj} = f'(net^h_{pj}) \sum_{k=1}^M (\delta_{pk} \cdot w^o_{kj}). \quad (5.10)$$

Basicamente, o processo de treinamento consiste de duas fases através das diferentes camadas da rede: propagação e adaptação. Na fase de propagação, um vetor de entrada (x_i) é apresentado na camada de entrada da rede (*input layer*), e o efeito se propaga através da rede, camada por camada (Fig. 5.5). O término desta fase se dá quando todos os neurônios da rede tiverem produzido suas respostas, gerando na camada de saída da rede (*output layer*) um vetor resposta (y_i). Nesta fase, o peso de cada sinapse da rede é mantido fixo.

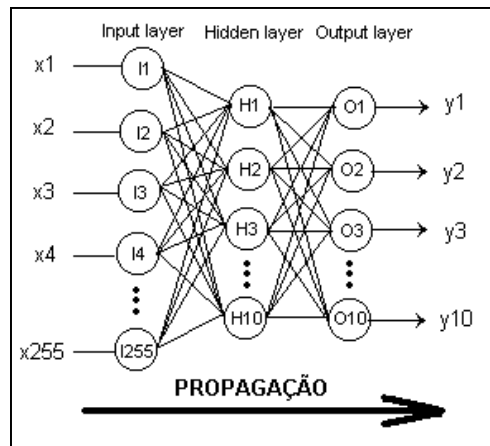


Fig. 5.5 – Fase de propagação

Durante a fase de adaptação, os pesos das sinapses são corrigidos segundo uma regra de correção por erro de acordo com as expressões deduzidas 5.7, 5.8, 5.9 e 5.10. Especificamente, a resposta atual da rede (y_i) é subtraída da resposta desejada (r_i) para produzir um sinal de erro. Este sinal é então propagado de volta através da rede em direção contrária às conexões sinápticas – por isso o nome *error back propagation*. Os pesos das sinapses são modificados para aproximar a resposta atual da rede da resposta desejada.

O principal objetivo deste procedimento é encontrar um conjunto de pesos que garantam que para cada vetor de entrada o vetor saída produzido pela rede é o mesmo (ou suficientemente próximo) do vetor de saída desejado (Fig.5.6).

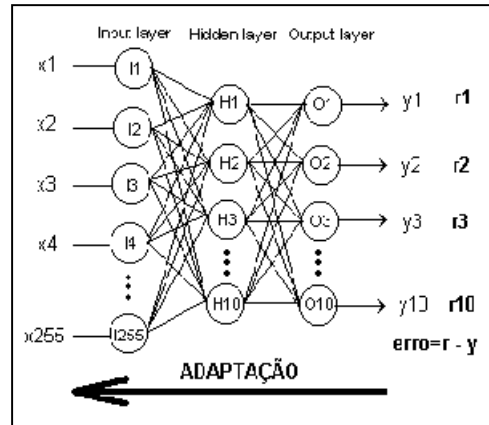


Fig. 5.6 – Fase de adaptação

Um processo de treinamento completo é composto das seguintes fases:

1) Inicialização aleatória dos pesos: Este é o primeiro passo para iniciar o treinamento da rede e deve ser executado uma única vez. Os valores dos pesos devem estar espalhados aleatoriamente dentro de valores pertencentes a faixa de atuação da função de ativação.

2) Propagação: Nesta fase o vetor de entrada é apresentado à camada de entrada da rede e uma resposta a este estímulo aparece na camada de saída.

3) Adaptação: Nesta fase é calculado o erro entre a resposta da rede e a resposta desejada e todos os pesos da rede são modificados de acordo com o erro apresentado.

4) Verificação do erro: Nesta fase é medido o erro apresentado pela rede e caso ele esteja abaixo de um valor aceitável, conforme desejo do projetista, o treinamento é encerrado; do contrário, um novo ciclo (propagação – adaptação - verificação) é executado.

CAPÍTULO 6

SISTEMA DE IDENTIFICAÇÃO AUTOMÁTICA DE VEÍCULOS

6.0 INTRODUÇÃO

Nos capítulos anteriores foram estudados, em separado, algumas técnicas de processamento, análise e interpretação de imagens. Neste capítulo, descrever-se-á com detalhes um sistema de identificação automática de veículos, suas principais aplicações, e apresentar-se-á o sistema, composto dos algoritmos vistos, desenvolvido neste trabalho.

6.1 INTRODUÇÃO AOS SISTEMAS DE IDENTIFICAÇÃO AUTOMÁTICA DE VEÍCULOS

Os engenheiros de tráfego necessitam adquirir dados sobre o trânsito para possibilitar seu estudo, controle e gerenciamento. Diversos tipos de sensores convencionais são utilizados para estes fins, tais como: *loops* magnéticos, sensores piezoelétricos, etc. Entretanto, as técnicas de processamento e análise de imagem possuem diversas vantagens sobre estes [16].

Dentre as diversas aplicações no trânsito, a identificação de veículos através da leitura de sua placa de licença vem conquistando cada vez mais espaço. No início dos anos cinquenta, este conceito era usado para estudar o tempo de duração de viagens entre origem e destino. Os primeiros métodos utilizados eram baseados em observadores que anotavam as placas dos veículos e os tempos correspondentes em um papel ou fita gravada. As placas eram manualmente comparadas mais tarde, e os tempos de viagem calculados [2]. O crescente avanço tecnológico tem aumentado substancialmente a precisão e facilidade desta técnica.

Quando um veículo entra no campo de visão de um SIAV (Sistema de Identificação Automática de Veículos), um complexo processo em cascata é iniciado. Um sensor, sensível à presença, informa ao computador a existência de um objeto de interesse. Este, por sua vez, aciona o sistema de aquisição para capturar a imagem, sincronizando a

iluminação, e adquire um quadro ou uma seqüência de quadros do veículo, armazenando-os.

Uma vez que a foto foi digitalizada, o próximo passo é verificar a existência da placa de licença e localizá-la na imagem. O sistema tem que procurar uma placa em um mar de objetos semelhantes, como adesivos nos pára-choques, etiquetas de fabricantes, etc.

Normalmente, vários testes são feitos para isolar a placa do restante da imagem, possibilitando a utilização de um sistema de reconhecimento de caracteres para sua leitura.

A Fig.6.1 mostra as partes físicas do sistema.

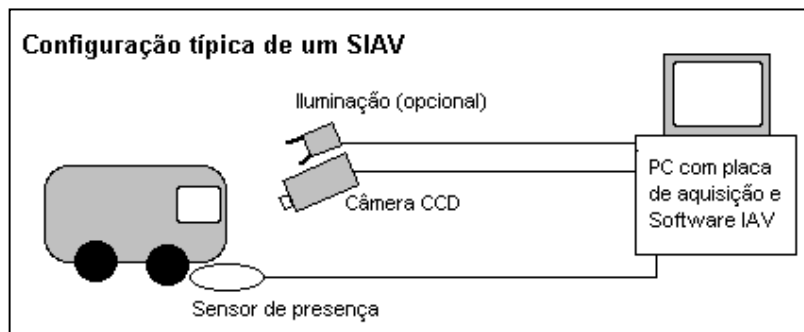


Fig. 6.1 – Configuração típica de um SIAV

6.2 APLICAÇÕES

As aplicações de um SIAV são muito variadas, neste item serão vistas as mais comuns.

Medição e monitoramento do tráfego

A medição e o monitoramento de várias características de viagens e fluxo de tráfego urbano, por meio da tecnologia de vídeo, vem sendo cada vez mais usados para o planejamento de transporte e gerenciamento de tráfego [7]. Estima-se que para o acompanhamento adequado do tráfego em uma auto-estrada é necessário que o sistema utilizado forneça sua resposta em menos de um segundo [17].

A possibilidade de se determinar padrões de movimento pela comparação de pares de placas de licença, ao longo de uma malha rodoviária, levou à primeira implementação de um sistema de identificação automática de veículos em 1990 na

Inglaterra. A praticidade deste tipo de sistema foi confirmada em uma série extensiva de tentativas conduzidas em 1993 pela *Volpe National Transportation Systems Center* (VMTSC) e seus associados [7].

As maiores vantagens deste tipo de sistema, para medição e monitoramento de tráfego, são:

- Possibilidade de fornecimento de grandes quantidades de amostras durante o período de coleta de dados;
- Possibilidade de fornecimento de uma estimativa representativa dos tempos de viagem através de amostragem aleatória;
- Fornecimento de tempos de viagem em pequenos intervalos de tempo, fornecendo um perfil de velocidade para estudo da seção durante os horários de pico;

Problemas de segurança

Neste item serão vistos alguns problemas de segurança que podem ser resolvidos com a utilização de um SIAV [18].

1) Estacionamento pago

A prática de fraudes em estacionamentos pagos traz importantes perdas para as companhias administradoras. Uma prática fraudulenta comum é o uso de dois carros, em ordem, para ocupar permanentemente um espaço no estacionamento.

Exemplo: Uma pessoa com um carro *A* entra no estacionamento em um instante de tempo T_1 , marcado no ticket de acesso TA . A mesma pessoa entra com um carro *B* em um instante de tempo T_2 , marcado em outro ticket TB . Assim, a pessoa pode deixar o carro *B* no estacionamento e sair com o carro *A* em um instante de tempo $T_2+\epsilon$, utilizando o ticket TB e pagando um valor mínimo referente ao tempo ϵ . Mais tarde, o carro *A* será guardado novamente no estacionamento com um ticket TA' , que será usado para retirar o carro *B*, novamente pagando apenas uma quantia mínima. A repetição deste processo traz grandes perdas para a companhia administradora, permitindo que uma pessoa mantenha um carro protegido no estacionamento pagando apenas um mínimo.

Outra típica situação é o furto de carros. Um ladrão entra em um estacionamento com seu próprio carro *A* obtendo um ticket *TA*. Ele rouba um veículo *B* e sai do estacionamento utilizando o mesmo ticket *TA*.

Este tipo de fraude traz enormes perdas às companhias administradoras tendo como consequência um aumento no custo das tarifas oferecidas.

Uma solução para os problemas apresentados é a utilização de um sistema que automaticamente reconhece a placa do veículo ao entrar no estacionamento, e grava a mesma no ticket de acesso. Mais tarde, quando o veículo for deixado no local, o sistema confere a informação presente no ticket com a placa de licença do veículo.

2) *Pagamento de pedágio*

Um sistema capaz de reconhecer placas de licença pode ser usado para identificar veículos em praças de pedágios. Isto pode ser feito de duas formas. Na primeira, o sistema pode ser usado, em conjunto com um banco de dados contendo os dados de registro do veículo e informações de seu proprietário, para debitar automaticamente a tarifa do pedágio. Este procedimento pode reduzir drasticamente os custos da concessionária da rodovia através da redução do número de pessoas necessárias no local, principalmente em momentos de tráfego intenso, diminuindo ainda o tempo de espera em postos de pedágio.

Na segunda forma este tipo de sistema pode ser usado como sistema de segurança objetivando identificar veículos fraudulentos. Por exemplo, na Itália, uma auto estrada controlada por uma companhia privada possui um sistema de identificação de veículos através de sensoriamento remoto chamado de “*Telepass*” [18]. Este sistema permite que carros portadores de um dispositivo especial transitem em uma via específica sem necessidade de parar nas praças de pedágio. Entretanto, motoristas fraudulentos podem tentar transitar por esta via evitando o pagamento de pedágio. Nestes casos, um sistema de identificação por imagem pode ser usado para coibir este tipo de comportamento.

3) Acesso a áreas restritas

O sistema pode ser usado para identificar o abuso em qualquer situação em que o tráfego é restrito. Por exemplo, os centros históricos de cidades como Roma e Florença são fechados para o tráfego público. Porém, muitas pessoas ignoram este fato e transitam por estes locais. Como estes centros possuem inúmeros pontos de acesso, a utilização de fiscais para o controle destes locais se torna inviável. Neste caso, um SIAV pode substituir a presença humana e ainda detectar alguns tipos de infração cometida.

6.3 ALGORITMOS DE LOCALIZAÇÃO E SEGMENTAÇÃO DOS SISTEMAS ESTUDADOS

Dentre os inúmeros trabalhos encontrados na literatura, convém destacar aqueles que possuem alguma influência sobre o sistema desenvolvido. São eles:

6.3.1 Identificação da placa através de análise por variação tonal

Os trabalhos de [9], [17], [18], [19] e outros, encontram-se nesta categoria. Todos baseiam-se no fato de que a área da placa de licença possui um padrão de variação tonal diferente do restante da imagem. Isto faz com que esta região possua uma “assinatura” padrão facilmente reconhecível, como pode ser visto na Figura 6.2.

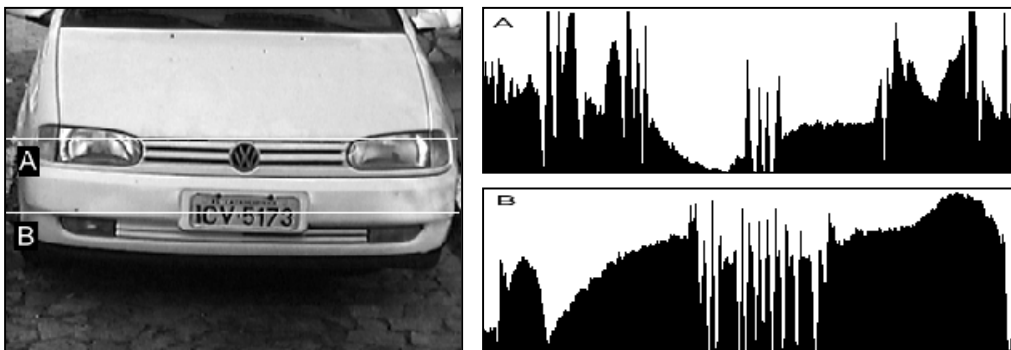


Fig. 6.2 – Imagem original e intensidades das linhas A e B

Ainda, uma vez que as placas de licença possuem sete caracteres e são fabricadas de acordo com leis governamentais, as transições de níveis de cinza entre caracteres e fundo constituem um tipo especial de padrão tonal que pode ser usado como objeto de procura. Uma breve descrição do algoritmo de Soh [19] será apresentada.

O método de Soh [19] é dividido em duas etapas:

(A) Extração das áreas candidatas a placa

Neste etapa, é feita uma procura por linhas horizontais da imagem que possuam um padrão de variação tonal esperado. Isso é feito em quatro fases:

- (1) Amostragem de linhas
- (2) Codificação
- (3) Agrupamento, e
- (4) Seleção de áreas candidatas

Fase 1: Amostragem de linhas

Para reduzir o tempo gasto na procura da placa, as linhas são amostradas em intervalos regulares τ . A escolha do valor de τ depende do tamanho dos caracteres procurados, devendo ser menor que a dimensão vertical dos mesmos.

Fase 2 : Codificação

Nesta fase é aplicado um operador unidimensional Sobel [6] sobre a linha amostrada. Os picos positivos representam as transições do fundo para os caracteres e os picos negativos o inverso. Para carros comerciais, a relação entre os picos positivos e as transições é ao contrário, uma vez que o fundo é mais escuro que os caracteres. Os picos menores que um valor θ e os picos negativos são descartados. Uma codificação em função da largura dos picos é feita, resultando em intervalos que representam as distâncias entre os elementos encontrados.

Fase 3 : Agrupamento

Nesta fase, os intervalos resultantes da fase anterior são analisados quanto a suas dimensões horizontais. Aqueles intervalos que estiverem dentro das dimensões esperadas são agrupados até formarem grupos com dimensões próximas as das placas. O resultado desta fase são inúmeros grupos que contém diferentes quantidades de intervalos válidos.

Fase 4 : Seleção das áreas candidatas

Os grupos formados na etapa anterior são analisados e, aqueles cujo número de intervalos estiverem entre o mínimo e o máximo número de intervalos possíveis para os caracteres existentes na placa, são aceitos como candidatos a área da placa.

(B) Verificação das áreas candidatas a placa

Devido a possibilidade de existirem diversas áreas candidatas, é necessário um modo de verificar qual delas é realmente válida. Este procedimento é feito através de uma procura bidirecional nas linhas adjacentes à linha amostrada. Quando um grupo de linhas é aproximadamente igual à dimensão vertical dos caracteres, este é considerado o local da placa.

Os outros métodos, como foi dito anteriormente, baseiam-se nos mesmos fundamentos, porém, não são apresentados detalhadamente em seus respectivos trabalhos.

6.3.2 Identificação da placa pela procura de elementos em uma imagem binarizada

O método de localização proposto por Botha [20], ao invés de basear-se em um padrão de variação tonal conhecido, analisa a mesma imagem, previamente binarizada, a procura de objetos de dimensões esperadas. O algoritmo de binarização executa a segmentação dos dígitos, sendo fundamental para o bom desempenho do sistema. Uma breve descrição do algoritmo de Botha [20] será apresentada a seguir.

O método proposto por Botha [20] pode ser dividido em quatro etapas:

(A) Pré-processamento de imagem

Neste etapa a imagem é binarizada através do algoritmo de binarização *Niblack* apresentado na seção 3.1. Utiliza-se uma janela de amostragem de 15x15 *pixels* e o valor α fixo em 0.2.

(B) Localização dos dígitos

Esta etapa é responsável pela localização de entidades alfanuméricas. Uma lógica itera através de todos os *pixels* da imagem e verifica, usando um conjunto de regras, se existe um candidato a dígito na posição corrente. Esta etapa consiste de dois passos:

(B.1) Procura por dígitos de tamanho conhecido

Neste passo assume-se que os dígitos procurados possuem dimensões aproximadamente fixas, o que pode ser considerado verdadeiro visto que a distância entre a câmera e o carro deve ser constante, e que o tamanho dos dígitos obedece a normas de controle. O algoritmo, propriamente dito, desloca uma estrutura retangular sobre a imagem, *pixel* a *pixel*, à procura de entidades que estejam completamente contidas em suas dimensões.

(B.2) Verificação do número de pixels

Os *pixels* pertencentes ao objeto encontrado no passo anterior são contados, e uma relação entre o número encontrado e o número de *pixels* pertencentes ao fundo é calculada. Se esta relação estiver acima de 15%, a posição encontrada é classificada como um potencial caracter alfanumérico, do contrário, ela é descartada.

(C) Localização final da placa

Nesta etapa, um retângulo de dimensões pré-definidas é colocado sobre os dígitos encontrados, verificando a quantidade de elementos inscritos nele. O retângulo que possuir o maior número de dígitos é considerado como provável local da placa.

6.4 O SISTEMA DESENVOLVIDO

O projeto de um sistema de visão artificial envolve diversas etapas e, conseqüentemente, diversas decisões. Estas decisões vão desde o número de bits necessários para representação da imagem até o método de reconhecimento que será empregado no sistema. Para que sejam coerentes durante o desenvolvimento de um sistema deste porte, diversas soluções devem ser exaustivamente testadas, o que demanda uma grande quantidade de tempo e trabalho.

No desenvolvimento deste projeto, os dois algoritmos apresentados anteriormente foram testados em um banco de imagens previamente selecionadas. O algoritmo proposto por Soh [19] mostrou-se muito rápido, porém deficiente em condições de iluminação adversas. Em contrapartida, o algoritmo proposto por Botha [20] mostrou-se eficiente no quesito variação de iluminação, porém, apresentou um tempo de processamento muito grande em relação à outra abordagem. Em vista dos resultados obtidos, a primeira decisão tomada foi de implementar um sistema que fizesse uso das vantagens apresentadas por cada um dos algoritmos estudados, evitando suas deficiências.

Serão apresentados, em ordem, todos os estágios que compõem o sistema desenvolvido.

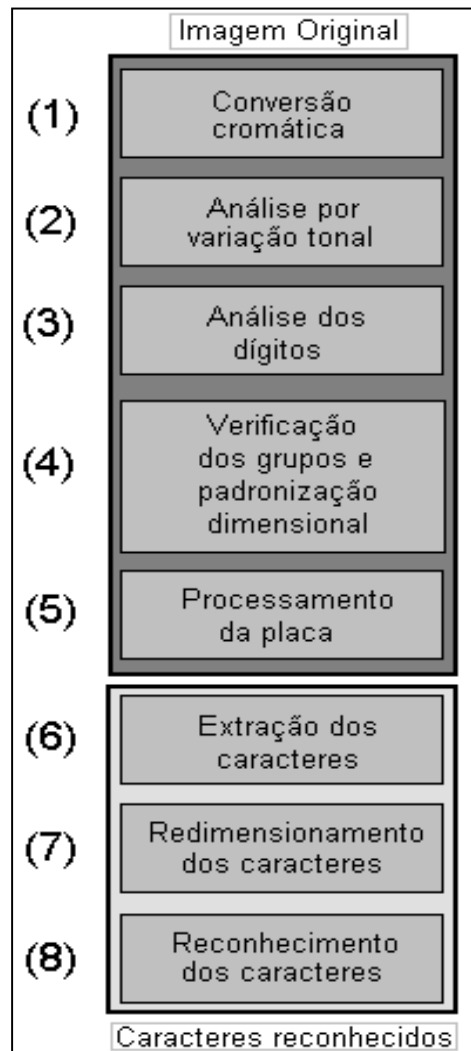


Fig. 6.3 – Etapas do sistema

As cinco primeiras etapas são responsáveis pela localização da placa dentro da imagem, e as três etapas restantes pela extração e reconhecimento dos caracteres nela contidos.

6.4.1 Conversão de uma imagem colorida para monocromática

A imagem adquirida pelo dispositivo de captura de vídeo é normalmente colorida e representada por 24 bits. Os algoritmos utilizados para localização e segmentação dos caracteres não fazem uso da informação de cor presente na imagem, pelo contrário, buscam padrões similares e descontinuidades entre *pixels* em uma imagem monocromática. Por estes motivos, se faz necessária uma conversão da imagem original colorida para o tipo monocromática (256 tons de cinza).

A conversão é feita através da análise dos valores das três componentes, vermelho, verde e azul, formadoras de cada *pixel* da imagem colorida. O *pixel* resultante $P'(x,y)$ na imagem monocromática pode ser calculado através da comparação dos valores das componentes de um *pixel* na imagem colorida, em busca dos valores mínimo e máximo, conforme a equação 6.1:

$$P'(x, y) = \frac{V_{\min} + V_{\max}}{2} . \quad (6.1)$$

Por exemplo: para um *pixel* pertencente à imagem original, cuja tríade seja (255,0,0), vermelho puro, o valor correspondente na imagem convertida $P'(x,y)$ será igual a média entre o maior e o menor valor.

$$P'(x, y) = \frac{V_{\min} + V_{\max}}{2} = \frac{0 + 255}{2} \cong 127$$

Esta conversão é o resultado parcial de uma conversão do sistema de cores RGB (*red, green, blue*) para o sistema HSL (*hue, saturation, luminance*). Ela é dita parcial porque fornece apenas uma das componentes do sistema HSL, a componente L de luminância. Esta componente é responsável pela informação de intensidade presente em uma imagem.

6.4.2 Análise por variação tonal

O algoritmo de análise por variação tonal faz uso de uma característica construtiva da placa, o contraste entre fundo e os caracteres, para localizá-la. Esta técnica baseia-se no fato de que as linhas onde a placa está localizada na imagem, possuem uma clara “assinatura” que faz usualmente possível distingui-las das linhas restantes da imagem [17].

Uma análise mais rigorosa do desempenho deste algoritmo permite concluir que, em casos onde a distribuição de luz sobre a superfície da placa não é homogênea, e ainda, em imagens onde há elementos complexos, como paralelepípedos, reflexos de árvores nos vidros, etc., há uma diminuição em sua eficiência. Portanto, uma modificação no algoritmo proposto por [19] é usada para localizar, de uma forma preliminar, o local da placa de licença na imagem. Este estágio é dividido em três etapas.

(1) Filtro Sobel:

Na primeira etapa, um filtro passa altas (Sobel) unidimensional é aplicado sobre cada linha amostrada da imagem original obtendo como resposta a variação horizontal do gradiente. Isto equivale a dizer que o operador é posicionado sobre cada *pixel* pertencente a linha analisada e seus coeficientes são multiplicados pelas respectivas intensidades da linha. Diversas combinações de valores foram testadas até se chegar aos valores finais que compõem o operador (Tabela 1).

0.1	0.25	0.75	-2.2	0.75	0.25	0.1
-----	------	------	------	------	------	-----

Tabela 1 – Coeficientes do operador Sobel

Este operador produz valores elevados nas bordas de elementos com alto contraste, enquanto que em elementos com pouco contraste sua resposta tende a se aproximar de zero. Portanto, os elementos que horizontalmente possuem alguma variação tonal aparecem, na saída do operador, com valor significativo.

Por ser um operador diferencial, seu resultado pode variar entre valores positivos e negativos, conforme o tipo de variação tonal apresentada: tons claros para tons escuros (positivo), tons escuros para tons claros (negativo).

Como explicado na seção 6.3.1 Fase 1, os valores negativos resultantes são desprezados, restando apenas as variações positivas na linha analisada. A figura 6.4 mostra o resultado do operador para uma imagem inteira.



Fig. 6.4 – Imagem original e saída do operador Sobel

No detalhe (Fig. 6.5) vemos a variação tonal da linha que cruza a área da placa da figura 6.4 e o resultado do operador Sobel, descartados os picos negativos, para a mesma linha. As variações dentro da área da placa destacam-se em relação ao restante dos elementos que compõem a linha.

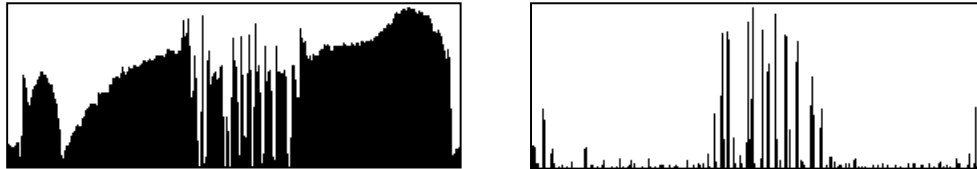


Fig. 6.5 – Detalhe da imagem original (linha) e saída do operador Sobel (picos positivos)

Convém lembrar que as linhas são amostradas em intervalos regulares τ , de baixo para cima, para acelerar o processo de procura, como mostra a fig. 6.6.



Fig. 6.6 – Sentido de varredura

(2) Análise das linhas

Uma vez que tenha sido aplicado o filtro Sobel na linha amostrada, uma seqüência de passos é executada para verificar a existência da placa. Os passos são:

(2.1) Marcação e análise dos picos encontrados

Todos os picos encontrados, independente de sua amplitude, são marcados em $x1$ e $x2$, indicando seu início e fim respectivamente (Fig. 6.7).

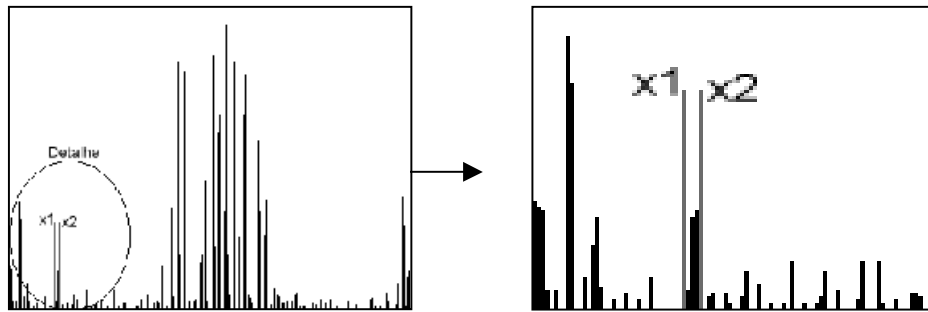


Fig. 6.7 – Resultado do filtro Sobel e detalhe da marcação dos picos

Aqueles picos cuja intensidade da imagem original nas marcas $x1$ e $x2$ não forem próximas, e ainda, cuja distância entre $x1$ e $x2$ for menor, ou maior, que a largura esperada de um segmento vertical de caracter são descartados. O resultado desta etapa são os picos que satisfazem as condições exigidas na equação 6.2.

$$\begin{aligned} |I(x1) - I(x2)| < \theta \\ v_{\min} < (x2 - x1) < v_{\max} \end{aligned} \quad (6.2)$$

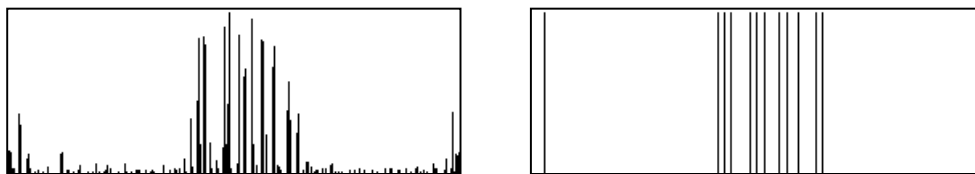


Fig. 6.8 – Resultado do filtro Sobel e da etapa de marcação e eliminação de picos

(2.2) Criação e análise de intervalos

O próximo passo é a criação de intervalos. Neste ponto as marcas $x1$ são descartadas e são criados grupos de intervalos entre as marcas $x2$ remanescentes. Uma análise dos grupos criados permite selecionar aquele que está dentro das dimensões horizontais

esperadas da placa, e que possui um número de intervalos, de tamanho pré-definido, entre um valor mínimo e máximo aceitáveis (Fig. 6.9).

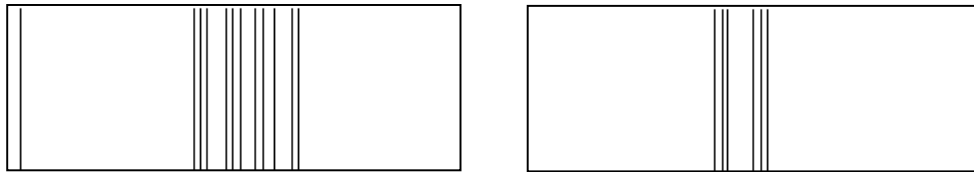


Fig. 6.9 – Resultado da etapa de marcação e eliminação de picos, e da criação e análise de intervalos

(3) Análise em linhas adjacentes

Uma vez encontrada uma linha que atenda aos quesitos da etapa anterior, uma nova análise é feita nas linhas adjacentes a ela. Quando um conjunto de linhas adjacentes maior que um tamanho pré-determinado, normalmente igual ao tamanho esperado do caracter, for encontrado, o algoritmo é terminado. Devido a algumas imperfeições na implementação deste algoritmo que causam uma imprecisão na localização exata do local da placa, e além disso, devido à grande variação dos tamanhos das placas contidas no banco de testes (entre 70x20 e 120x40 *pixels*), o local candidato a placa é inscrito em um retângulo de dimensões fixas, 120x50 *pixels*, aumentando a chance da placa estar contida nesta região, conforme pode-se observar na figura 6.10.

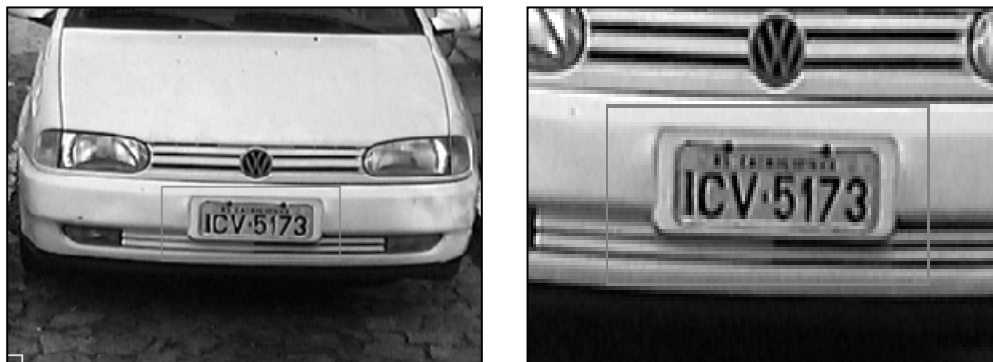


Fig. 6.10 – Resultado final do algoritmo de análise por variação tonal e detalhe

O algoritmo de análise por variação tonal pode ter como resultado uma das três seguintes saídas:

- (1) O local correto da placa - assinalado por um retângulo de dimensões 120x50 *pixels*;
- (2) Um local incorreto - assinalado por um retângulo de dimensões 120x50 *pixels*;
- (3) Nenhum local encontrado - nenhuma marca na imagem;

6.4.3 Análise dos dígitos

O algoritmo proposto por Botha [20] mostrou-se mais eficiente do que o algoritmo de análise por variação tonal em relação aos problemas de iluminação. Porém, testes mais exigentes mostraram que seu desempenho é muito dependente de como a luz está distribuída sobre a superfície da placa. Não obstante, outro problema em relação a sua implementação é o tempo de processamento. Levando-se em conta as deficiências apresentadas, o autor desenvolveu um algoritmo adaptativo para localização da placa. O algoritmo desenvolvido é dividido em três fases (Fig. 6.11).

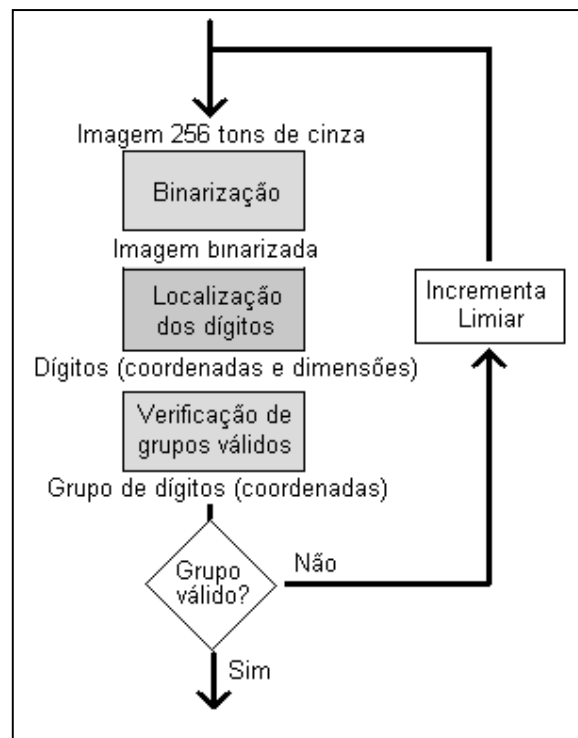


Fig. 6.11 – Fases da análise de dígitos

6.4.3.1 Binarização *Niblack* modificada

Como explicado da seção 4.1.3, a binarização *Niblack* é um algoritmo de binarização local adaptativa muito eficiente [21]. Para cada *pixel* da imagem é calculada a média e o desvio padrão em torno de sua vizinhança. Uma imagem de 320x240 *pixels* com 8 bits por *pixel* demora aproximadamente 9s para ser processada em uma plataforma PC com 400 MHz e 64 MB RAM.

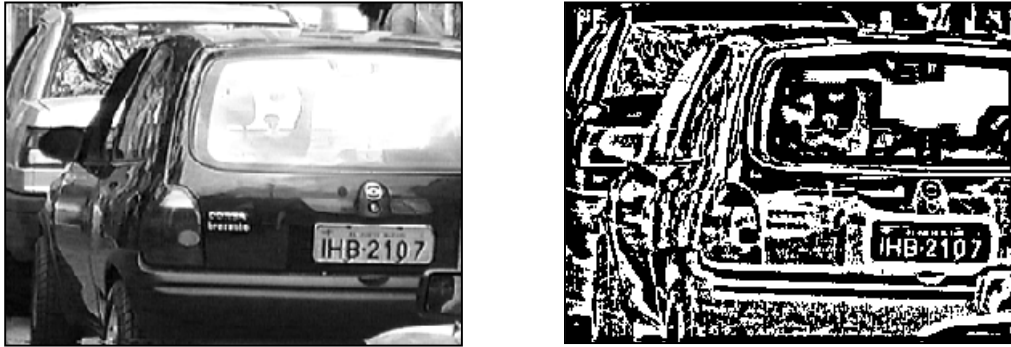


Fig. 6.12 – Resultado do algoritmo de binarização *Niblack*

Como se está interessado em minimizar o tempo de processamento do sistema, uma alteração neste algoritmo se faz necessária. Ao invés de se trabalhar com todos os 225 *pixels* da janela de varredura, optou-se por diminuir o número de *pixels* analisados para 49 (Fig. 6.13), alternando suas posições. Com esta janela de varredura o tempo de processamento cai para 2 segundos sem perdas significativas de qualidade da imagem resultante.

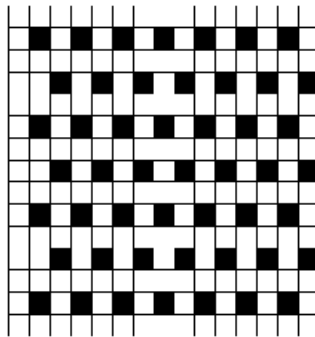


Fig. 6.13 – Janela de varredura

6.4.3.2 Localização dos dígitos

Uma vez que a imagem tenha sido binarizada pela etapa anterior, a localização dos dígitos é feita em três estágios (Fig. 6.14).



Fig. 6.14 – Algoritmo de localização de dígitos

Estágio 1: No primeiro estágio, a imagem binarizada é analisada a procura de uma região, com o formato de um L invertido, cujos *pixels* nela contidos estejam todos em branco. As dimensões do L invertido devem corresponder às dimensões dos caracteres esperadas.



Fig. 6.15 – Algoritmo de localização de dígitos – estágio 1

Estágio 2: Encontrada uma posição que satisfaça as condições do estágio 1, uma barra horizontal é deslocada, de cima para baixo da região, até encontrar o início e o término do elemento.



Fig. 6.16 – Algoritmo de localização de dígitos – estágio 2

Estágio 3: Se a dimensão vertical do elemento estiver de acordo com a dimensão vertical dos caracteres esperados, uma barra vertical é deslocada, da esquerda para a direita da região, até encontrar o início e o término do elemento.



Fig. 6.17 – Algoritmo de localização de dígitos – estágio 3

Os elementos cujas dimensões horizontais e verticais encontradas estiverem dentro de uma faixa esperada são armazenados em uma lista juntamente com seu ponto de referência (âncora). Estes três parâmetros são suficientes para identificar cada elemento e localizá-lo na imagem (Fig. 6.18).

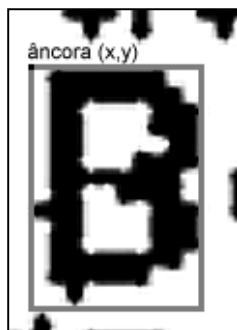


Fig. 6.18 – Ponto de referência (âncora)

6.4.3.3 Verificação de grupos válidos

O algoritmo de verificação de grupos válidos consiste em analisar cada um dos dígitos encontrados verificando a possibilidade de formação de grupos que possuam características em comum, sendo elas:

- O grupo deve possuir no mínimo 6 elementos e no máximo 7;

O sistema espera encontrar ao menos 6 caracteres. É intuitivo que as chances de se encontrar 6 caracteres em uma placa são maiores que as chances de se encontrar todos os 7 caracteres. Quando o sistema encontra apenas seis caracteres da placa ele procura inferir a posição do sétimo caracter em função da posição dos outros caracteres encontrados. A restrição quanto ao número máximo de caracteres diminui as chances do sistema em se confundir com possíveis adesivos, marcas de veículos, etc.

- A distância entre o primeiro e o último elemento não pode exceder a dimensão horizontal esperada da placa;

Com este procedimento espera-se restringir a formação de grupos que não satisfazem as dimensões horizontais esperadas. A soma das dimensões horizontais dos 7 caracteres deve ser menor que a dimensão horizontal da placa.

- A diferença entre a dimensão vertical dos elementos de um mesmo grupo não pode ser maior que 4 *pixels*;

A dimensão vertical dos caracteres de um placa é fixa, ao contrário da dimensão horizontal. Este procedimento visa eliminar outros objetos que se assemelhem aos caracteres, porém não possuam uma dimensão vertical próxima dos mesmos.

- A diferença entre a distância vertical de cada elemento não pode ser superior a 4 *pixels* em relação ao elemento anterior;

Este procedimento permite que o sistema encontre placas levemente inclinadas, uma vez que nestes casos cada caracter estará acima, ou abaixo, do seu caracter vizinho.

Normalmente, como resultado desta etapa, tem-se um único grupo que satisfaz todas as condições exigidas. No caso de existir mais de um grupo válido, aquele que possuir a menor diferença vertical entre as âncoras de seus elementos será considerado o grupo vencedor.

6.4.4 Verificação do número de caracteres e padronização dimensional

Nesta etapa o sistema verifica se o grupo vencedor possui 6 ou 7 caracteres e padroniza suas dimensões se necessário. No caso do grupo possuir 7 caracteres, ou 6 caracteres sendo o caracter que falta não pertencente às extremidades da placa (Fig.6.19) , o primeiro passo desta etapa não é executado.



Fig. 6.19 – 6 caracteres encontrados

(1) Localização do sétimo caracter

O primeiro passo para se tentar inferir a posição do caracter que falta é o cálculo das dimensões médias do grupo.

$$alt_{med} = \frac{1}{N} \sum_{c=0}^N alt_c \quad lar_{med} = \frac{1}{N} \sum_{c=0}^N lar_c \quad (6.4)$$

Onde N é igual ao número de caracteres, 6 neste caso.

O passo seguinte é descobrir entre quais caracteres encontra-se a maior distância entre os caracteres encontrados. Este provavelmente será o local onde fica o hífen da placa e permitirá inferir o provável local do caracter que falta: extremidade esquerda ou extremidade direita. No detalhe da figura (Fig. 6.20) pode-se observar que a maior distância entre os caracteres encontrados está no intervalo B. Assim, conta-se quantos caracteres encontram-se a direita do intervalo, se o número de caracteres for menor que 4 o sétimo caracter provavelmente encontra-se na extremidade direita da placa. Ao contrário, se o número de caracteres à direita do hífen for igual a 4, então o sétimo caracter provavelmente encontra-se na extremidade esquerda da placa.

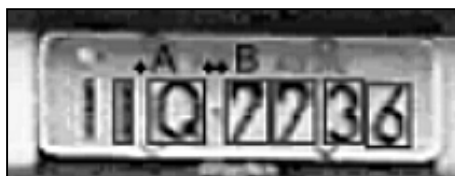


Fig. 6.20 – Intervalos (A>B)

As dimensões do sétimo caracter são iguais às dimensões horizontal e vertical médias calculadas em (6.4) acrescidas de um *pixel* cada. A posição (x,y) da âncora do novo caracter é calculada da seguinte forma:

Posição x: Um intervalo entre a posição dos dois caracteres que o antecedem ou o sucedem é calculado. Este intervalo servirá para posicionar o novo caracter. No caso da figura 6.20, o intervalo A, entre as letras I e Q, servirá como intervalo entre o novo caracter e o seu caracter sucessor.

Posição y: A posição em y é igual a posição do seu elemento adjacente acrescida de um *pixel*, teste práticos comprovaram que este *pixel* auxilia na etapa de extração dos caracteres.

A figura 6.21 mostra o resultado desta fase, em cinza o novo caracter e suas dimensões estimadas:



Fig. 6.21 – Novo caracter (NC)

(2) Padronização dimensional

A padronização dimensional ocorre naqueles elementos que possuem dimensão, vertical e/ou horizontal, inferior ou superior à média calculada. Neste caso, suas dimensões são alteradas para igualar-se à média dos caracteres presentes.

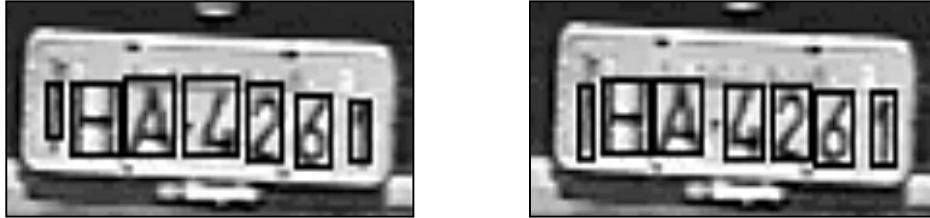


Fig. 6.22 – Padronização dimensional

6.4.5 Processamento da região da placa

Nos casos em que a informação contida na placa não se encontra facilmente reconhecível, é necessário um procedimento auxiliar que permita facilitar o processo de segmentação dos caracteres. A técnica de *contrast stretching* explicada na seção 3.1.2 é uma boa solução para ressaltar as diferenças entre caracteres e fundo da placa, porém requer um cuidado especial em sua utilização. Como se trata de uma técnica que expande o histograma da imagem, é necessário que a imagem a ser processada possua um histograma compactado em uma região do espectro de intensidade luminosa, do contrário a técnica não surtirá efeito algum. Esta necessidade torna sua utilização impraticável quando se trabalha com imagens adquiridas em ambientes sem controle de iluminação.

Para superar esta deficiência, Comelli [9] sugere uma alteração na técnica para sua utilização restrita à área da placa. Ao invés de procurar o menor e o maior tom de cinza do histograma da imagem e calcular o coeficiente de expansão em função destes valores, ele sugere a procura de tons de cinza de acordo com as equações (6.4).

$$\sum_{i=0}^a h(i) \geq \frac{nm}{10}, \sum_{i=0}^b h(i) \leq \frac{9nm}{10} \quad (6.4)$$

Onde n e m são as dimensões da área da placa, $h(i)$ a função histograma da mesma, e a e b correspondem aos valores de tons de cinza.

A técnica proposta por Comelli [9] é eficaz mesmo que o histograma não esteja compactado em uma região do espectro de tons de cinza. Assim, nesta etapa, uma

região da imagem é recortada entre a âncora do caracter mais a esquerda da placa e a âncora mais a direita do último caracter, acrescidos de 10 *pixels* na horizontal e 5 *pixels* na vertical (Fig.6.23).



Fig. 6.23 – Área de atuação do contrast stretching

Esta área é submetida ao processamento explicado na seção 3.1.2 e como resultado obtêm-se uma imagem com um maior contraste entre os elementos de interesse e o fundo, como pode ser visto nas figuras 6.24 e 6.25.

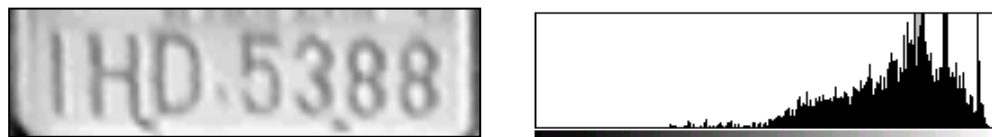


Fig. 6.24 – Imagem original e histograma compactado

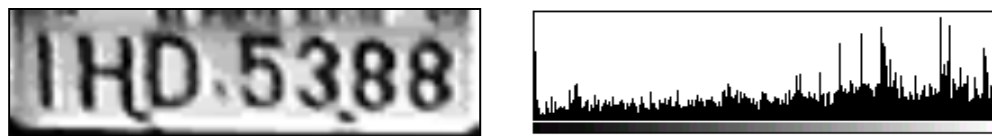


Fig. 6.25 – Resultado do contrast stretching e histograma expandido

Em placas cujo contraste já é adequado não há uma melhora significativa (Fig. 6.26). As características originais são preservadas, permitindo uma segmentação adequada dos caracteres.

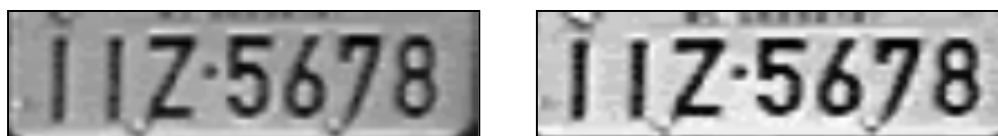


Fig. 6.26 – Imagem original e resultado do contrast stretching

O algoritmo de análise dos dígitos é novamente aplicado, porém, desta vez, apenas sobre a nova imagem realçada.

6.4.6 Extração dos caracteres encontrados

Uma vez que o local da placa tenha sido encontrado, realçado, e definidas suas dimensões finais, é necessário extrair os caracteres do restante da imagem. Isto é feito

através da utilização de um algoritmo de agrupamento por similaridade, que tem a finalidade de separar cada elemento encontrado, e de um algoritmo de avaliação dos elementos extraídos, para descartar possíveis ruídos indesejáveis. Estes dois algoritmos foram desenvolvidos pelo autor para esta aplicação e são descritos a seguir:

(1) *Agrupamento por similaridade:*

O algoritmo de segmentação por agrupamento de características similares efetua uma busca em uma imagem previamente binarizada a procura de quaisquer elementos diferentes do fundo. A busca é feita da esquerda para a direita, e de baixo para cima da imagem. Quando um *pixel* diferente do fundo é encontrado o algoritmo guarda sua posição e a de todos seus *pixels* vizinhos.

A vizinhança de cada *pixel* é analisada, conforme a ordem mostrada na figura 6.27, até que não exista mais nenhum *pixel* vizinho que não tenha sido analisado. Terminada a análise de todos os *pixels* daquele elemento, seus *pixels* são armazenados em uma lista e o mesmo é recortado da imagem. A imagem é preenchida com a cor do fundo em seu lugar (Fig. 6.28), e o algoritmo prossegue sua função a partir da posição do primeiro *pixel* do elemento encontrado.

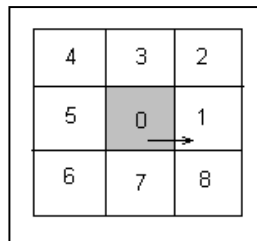


Fig. 6.27 – Orientação da procura dos *pixels* vizinhos

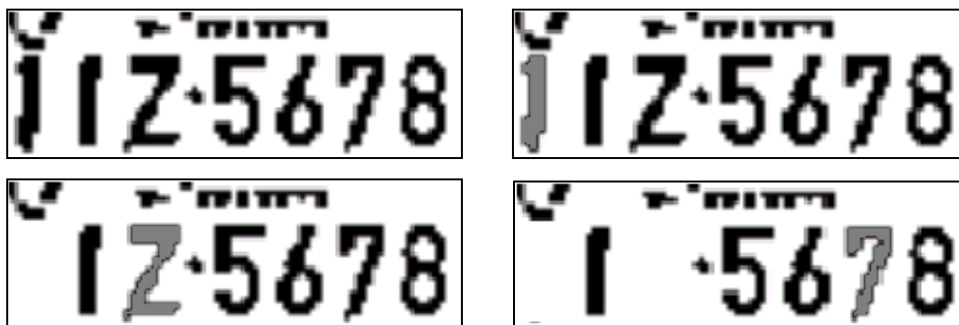


Fig. 6.28 – Etapas da segmentação por similaridade

(2) *Análise dos elementos encontrados:*

Nesta etapa, os elementos cujo número de *pixels*, contidos dentro do retângulo que inscreve os caracteres encontrados, for maior que o número de *pixels* da dimensão vertical média dos caracteres são considerados válidos. Os elementos que estiverem entre caracteres válidos, e que possuírem um número de *pixels* maior que 20% do tamanho médio dos caracteres também são considerados válidos. Todos os elementos considerados válidos são ordenados conforme sua posição horizontal, de forma a possibilitar a utilização de duas redes neurais distintas para seu reconhecimento, uma para as letras (3 primeiros caracteres) e outra para os algarismos (4 caracteres restantes).

Na seqüência de figuras abaixo fica claro o procedimento de análise dos elementos encontrados. O ruído acima do primeiro caracter (letra I – Fig. 6.30) é rejeitado por não estar inscrito em nenhum dos retângulos representantes da área dos caracteres encontrados. O ruído entre os caracteres Z e 5 (Fig. 6.30), apesar de estar entre dois caracteres, possui um número de *pixels* inferior ao limite de 20% da dimensão média dos caracteres. Na figura 6.31 os caracteres selecionados aparecem em ordem de apresentação às redes neurais.



Fig. 6.29 – Caracteres inscritos e suas dimensões

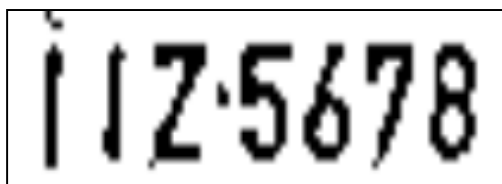


Fig. 6.30 – Caracteres e ruídos provenientes da etapa de binarização

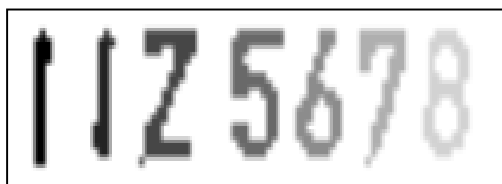


Fig. 6.31 – Resultado da análise dos elementos encontrados

6.4.7 Redimensionamento dos caracteres selecionados

Os elementos encontrados e analisados com sucesso na etapa anterior devem ser redimensionados para apresentação ao sistema de reconhecimento baseado em redes neurais. Como mencionado anteriormente, as redes neurais *feedforward* possuem um número fixo de neurônios na camada de entrada. Por este motivo, é necessário que o vetor de entrada possua uma dimensão fixa, igual ao número de neurônios da camada de entrada para o correto funcionamento da rede. Os caracteres são redimensionados através de um coeficiente de proporção vertical dimensional, ou seja, é calculado um coeficiente entre a altura do caracter encontrado e a dimensão vertical utilizada na rede (15 *pixels*). A dimensão vertical do caracter é alterada para 15 *pixels* e a dimensão horizontal é modificada conforme o coeficiente calculado.

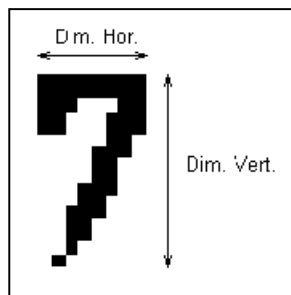


Fig. 6.32 – Dimensões dos caracteres

Por exemplo, um caractere de dimensões 10x20 *pixels* possui um fator de proporção $20/15 \approx 1.33$, a altura do caractere redimensionado será de 15 *pixels* e a largura ficará em $10/1.33 \approx 8$ *pixels*. As figuras 6.33 e 6.34 mostram um exemplo real de redimensionamento.



Fig. 6.33 – Caracteres selecionados sem dimensão fixa

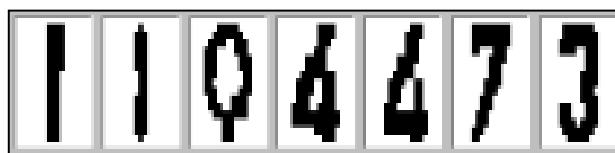


Fig. 6.34 – Caracteres redimensionados para matriz 15x15 *pixels*

Este procedimento garante que o caracter não excederá o número de neurônios na camada de entrada, uma vez que a altura dos caracteres das placas dos veículos é na maioria dos casos maior que a largura dos mesmos.

6.4.8 Reconhecimento dos caracteres

Uma vez que os caracteres encontrem-se redimensionados, os três primeiros são apresentados à rede responsável pelo reconhecimento das letras, e os quatro caracteres restantes são apresentados a rede responsável pelo reconhecimento dos números. Ambas as redes possuem 255 neurônios na camada de entrada, o que corresponde a um caracter de dimensões 15x15 *pixels* mais as duas linhas de descrição apresentadas no capítulo 4 (Fig. 6.35).

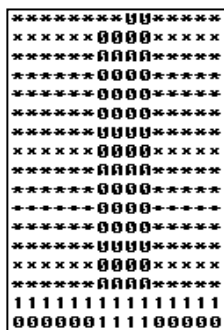


Fig. 6.35 – Caracter I (15x15 *pixels*)

A rede dedicada ao reconhecimento dos números possui 10 neurônios na camada escondida e 10 neurônios na camada de saída. A rede para reconhecimento de letras possui 26 neurônios nas camadas escondida e de saída (Fig. 6.36).

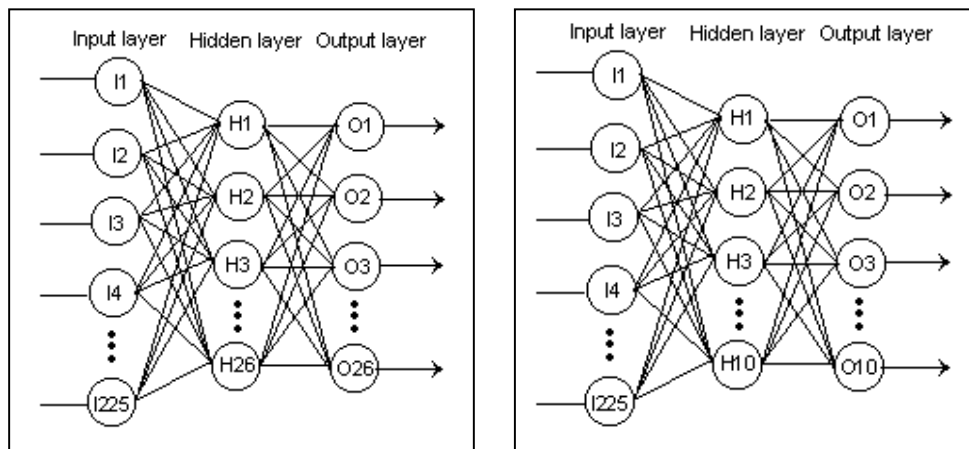


Fig. 6.36 – Redes neurais para letras e números

A saída da rede rotula o padrão de entrada com o neurônio de resposta mais elevada. Em outras palavras, uma vez que um determinado vetor de entrada é apresentado à rede, o neurônio da camada de saída que possuir a maior resposta indicará a que classe aquele vetor pertence. No caso da rede dedicada aos números, a associação do número reconhecido é diretamente ligada ao neurônio vencedor (neurônios de 0 a 9). Na rede para reconhecimento das letras, cada neurônio da camada de saída está associado a uma letra do alfabeto como pode ser visto na tabela abaixo.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Tabela 2 – Associação dos neurônios da camada de saída

A rede neural dedicada ao reconhecimento dos números foi treinada com 15 amostras de cada caracter, sendo estas selecionadas entre um banco de imagens específico para este fim. O processo de treinamento consistiu na apresentação das 15 amostras de cada caracter, um caracter por vez, até a rede convergir para um erro quadrático médio menor que 0,005.

Devido ao maior número de classes da rede neural dedicada ao reconhecimento das letras, e o tempo restante para o término deste trabalho, esta rede foi treinada com apenas 5 amostras para cada letra, obedecendo o mesmo processo descrito acima.

Duas outras redes neurais foram treinadas sem a utilização da descrição dos caracteres descrita no capítulo 4, porém os resultados foram inferiores aos apresentados pela rede que utiliza a informação da descrição.

CAPÍTULO 7

RESULTADOS

7.0 INTRODUÇÃO

Neste capítulo são apresentados os resultados obtidos durante as principais etapas de processamento, análise e reconhecimento das imagens. Para verificar a eficiência dos algoritmos utilizados no desenvolvimento do sistema utilizou-se um conjunto de 500 imagens adquiridas em diferentes oportunidades, com diferentes equipamentos e características. Devido ao grande número de imagens, apenas algumas delas são mostradas em detalhes. Uma comparação do desempenho do sistema em relação a um *software* demonstrativo, desenvolvido por uma empresa israelense [3], disponível na Internet é apresentada ao final.

Um computador PC AMD K6-2 400 MHz com 64 MBRAM foi utilizado para o processamento das imagens, e, portanto, as informações referentes ao tempo de processamento neste capítulo são para esta plataforma.

7.1 ANÁLISE POR VARIAÇÃO TONAL

Como mencionado anteriormente, este algoritmo possui um bom desempenho em imagens que possuam uma distribuição de luz homogênea sobre a superfície da placa do veículo. Nos casos em que isto não ocorre, ou em casos em que existam elementos de grande variação tonal (paralelepípedos, reflexos de árvores em vidro, etc), o algoritmo não consegue encontrar corretamente o local da placa. Os parâmetros utilizados no algoritmo são $\tau = 5$, $\theta = 25$, $v_{\min} = 1$ e $v_{\max} = 5$.

As imagens apresentadas a seguir mostram as etapas da aplicação do filtro Sobel, a marcação dos intervalos e o resultado final assinalado pelo retângulo de 120x50 *pixels*. Convém lembrar que as etapas de filtragem e marcação dos intervalos ocorrem somente naquelas linhas da imagem que satisfazem as condições impostas pelo algoritmo, e que as imagens mostradas servem apenas para permitir um melhor entendimento do seu funcionamento.

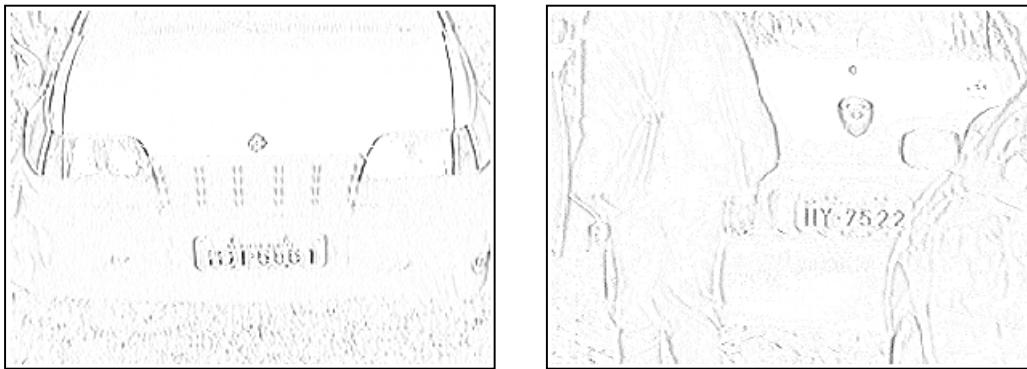


Fig. 7.1 - Operador Sobel

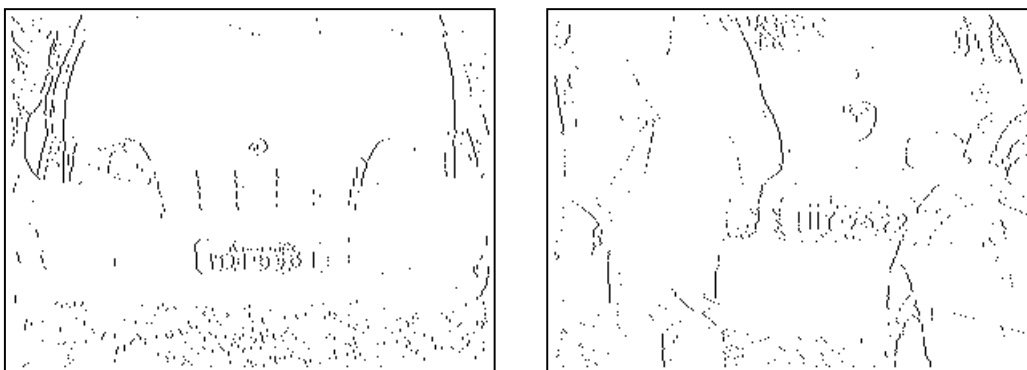


Fig. 7.2 - Marcação dos intervalos



Fig. 7.3 - Resultado final

Na seqüência de imagens mostrada na coluna da esquerda vê-se que o algoritmo confunde-se com as variações de tom dos paralelepípedos no solo, enquanto que na seqüência de imagens da coluna da direita o algoritmo consegue localizar corretamente a placa do veículo, apesar da aparente complexidade da imagem.

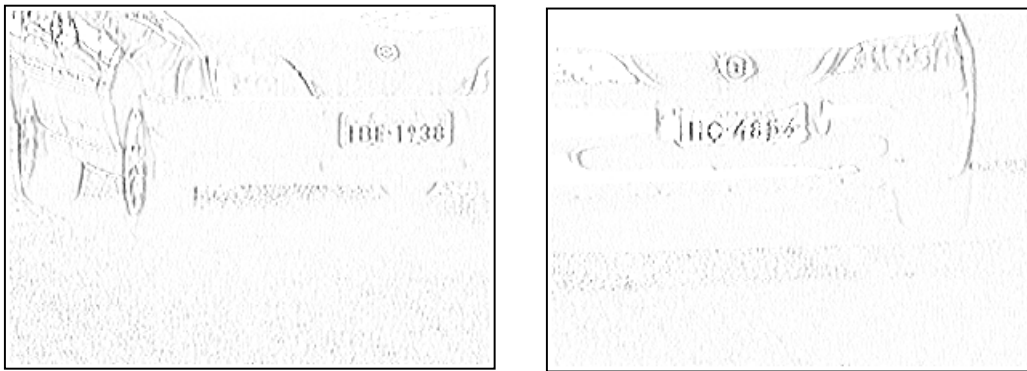


Fig. 7.4 - Operador Sobel

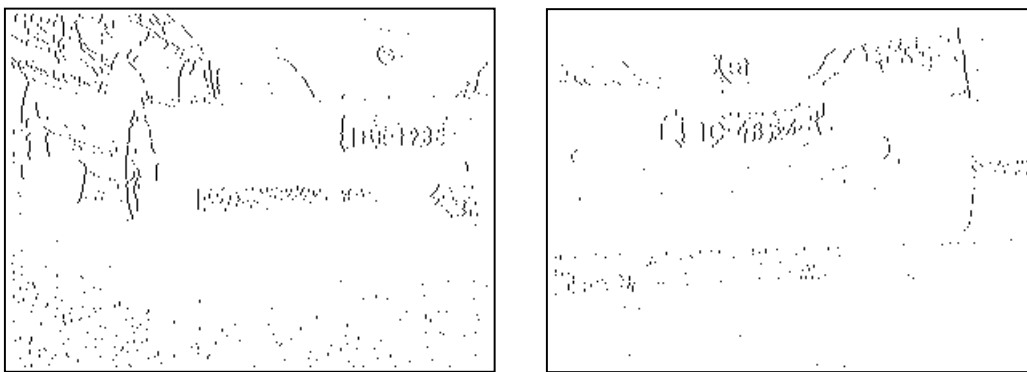


Fig. 7.5 - Marcação dos intervalos

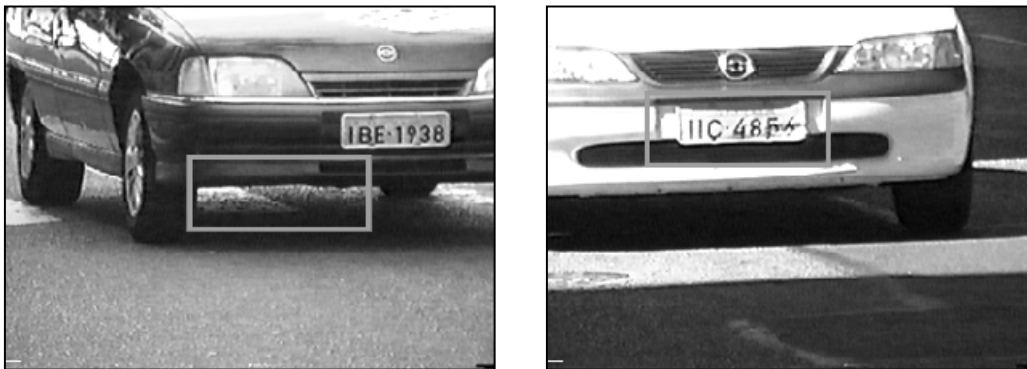


Fig. 7.6 - Resultado final

Novamente, na seqüência de imagens da coluna da esquerda, vê-se que o algoritmo confunde-se com as variações de tom no asfalto como mostra claramente o retângulo no centro da última imagem. Na seqüência de imagens da coluna da direita vê-se que somente o local da placa apresenta condições de ser considerado local válido pelo algoritmo, apesar da aparente dificuldade de leitura da placa.

O algoritmo de análise tonal foi testado em 500 diferentes imagens e apresentou um desempenho de 83,6 % de sucesso na localização da placa com um tempo médio de processamento de 0.03 segundos.

7.2 ANÁLISE POR DÍGITOS

O algoritmo de análise por dígitos utiliza-se do algoritmo anterior para acelerar o processo de procura da placa. Nos casos em que o algoritmo de análise tonal obtém sucesso, o tempo de processamento é reduzido, pois assim o algoritmo de análise por dígitos fica limitado a uma imagem de 120x50 *pixels*; caso contrário, toda a imagem é analisada consumindo mais tempo.

Este algoritmo é adaptativo e varia o valor do limiar α , iniciando em 0.2 e incrementando de 0.1 até chegar ao valor limite de 1. Este procedimento de variação de um limiar dentro de um algoritmo que já é adaptativo por natureza, uma vez que leva em conta a vizinhança em torno de cada *pixel*, confere uma robustez maior ao sistema permitindo uma performance elevada mesmo em situações adversas de iluminação.

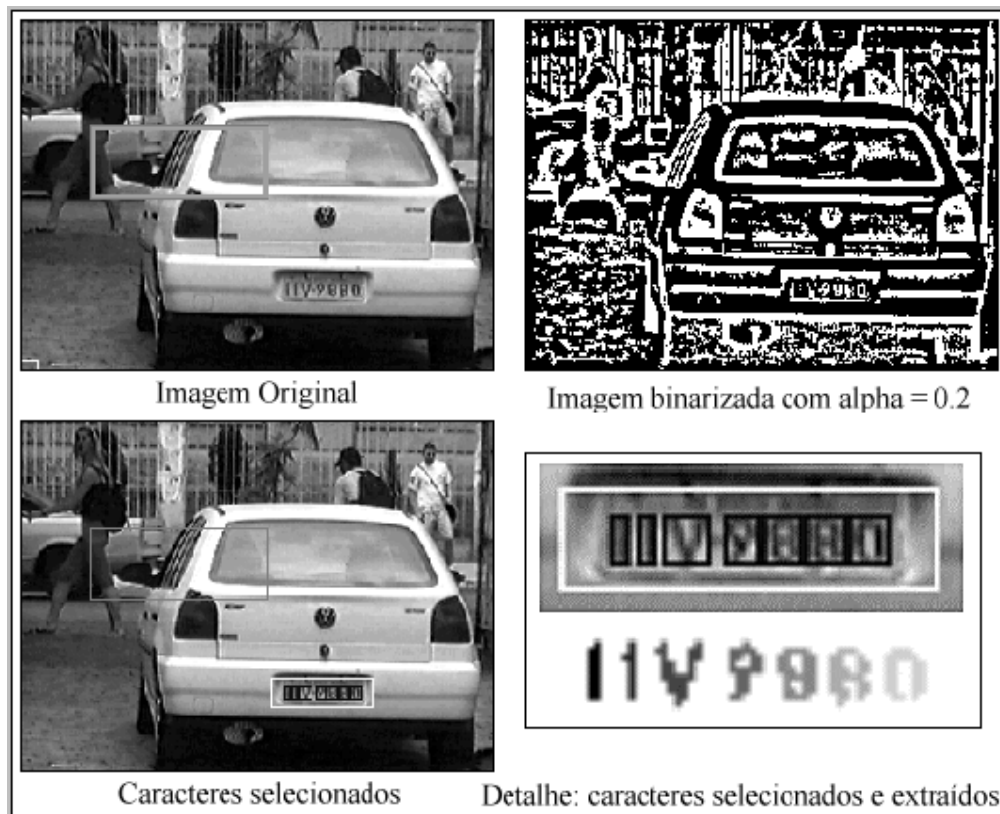


Fig. 7.7 – Etapas do sistema

Na figura 7.7 têm-se um caso em que o algoritmo de localização por análise tonal não obteve sucesso, informando um local errado para a placa. Neste caso, o algoritmo de análise por dígitos verifica se no local informado (retângulo 120x50 *pixels*) existe alguma placa, através do processo de binarização - localização dos dígitos - verificação de grupos válidos, variando o valor do limiar α sempre que necessário. No caso de não ser encontrado um grupo válido de dígitos neste local, o restante da imagem é analisada. O tempo de processamento total desta imagem é de 18 segundos e a resposta da rede neural é IAY9080 (a resposta correta é IIV9880).

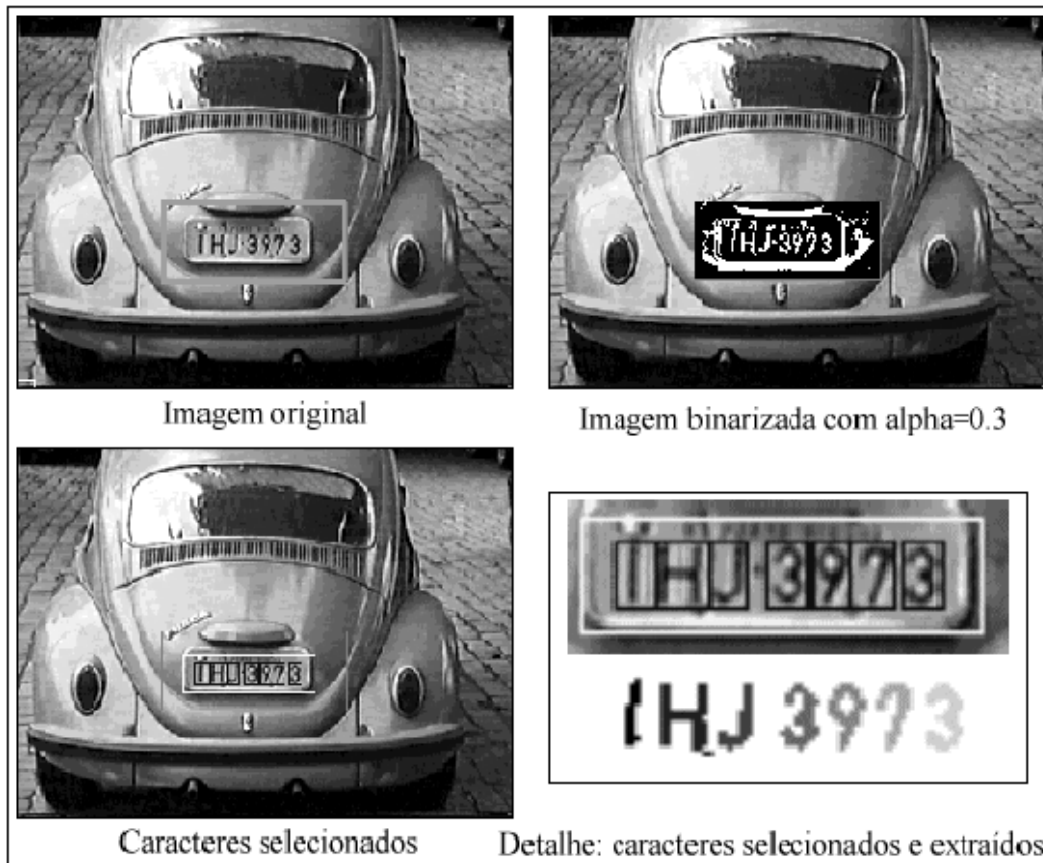


Fig. 7.8 – Etapas do sistema

Na figura 7.8 o algoritmo de localização por análise tonal obteve sucesso. Neste caso, o algoritmo de análise por dígitos se restringe apenas ao local indicado (retângulo 120x50 *pixels*) localizando corretamente os dígitos. O tempo de processamento para esta imagem fica reduzido a 3 segundos e a resposta da rede neural é IHJ1973 (a resposta correta é IHJ3973).

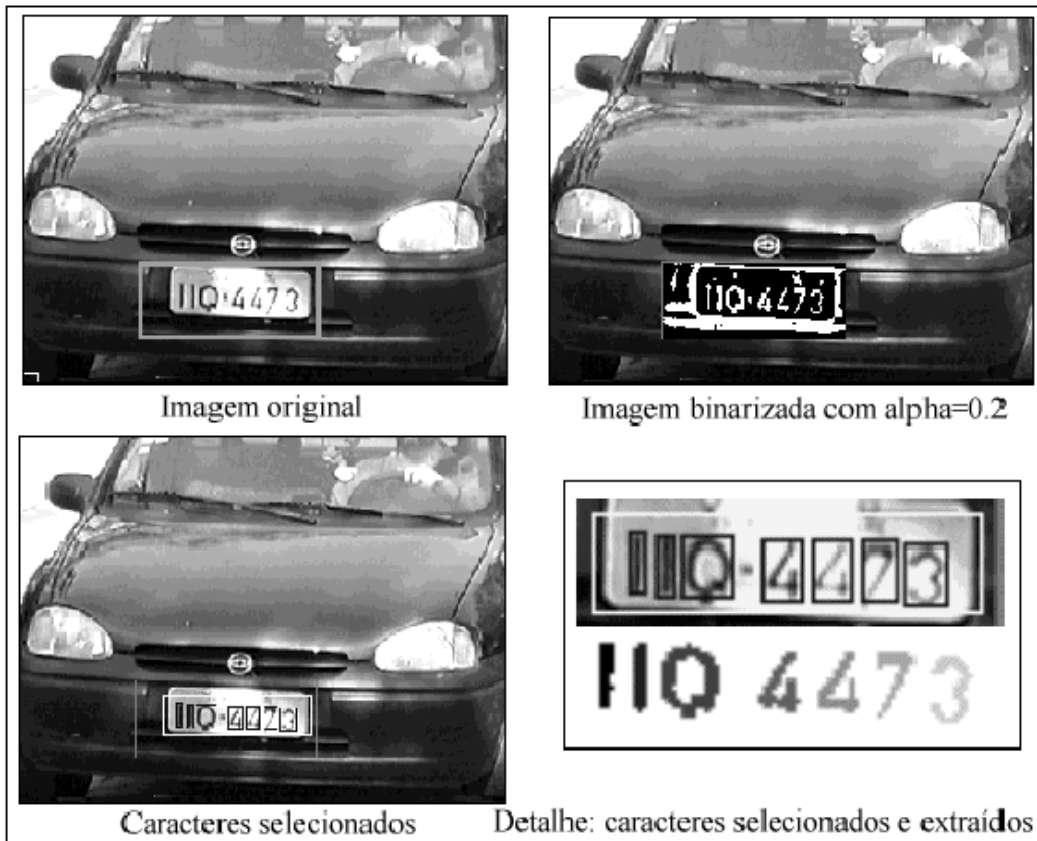


Fig. 7.9 – Etapas do sistema

Na figura 7.9 o algoritmo de localização por análise tonal também obteve sucesso. Novamente, o algoritmo de análise por dígitos se restringe apenas ao local indicado localizando corretamente os dígitos. O tempo de processamento para esta imagem fica reduzido a 2.7 segundos e a resposta da rede é IIO4473 (a resposta correta é IIO4473).

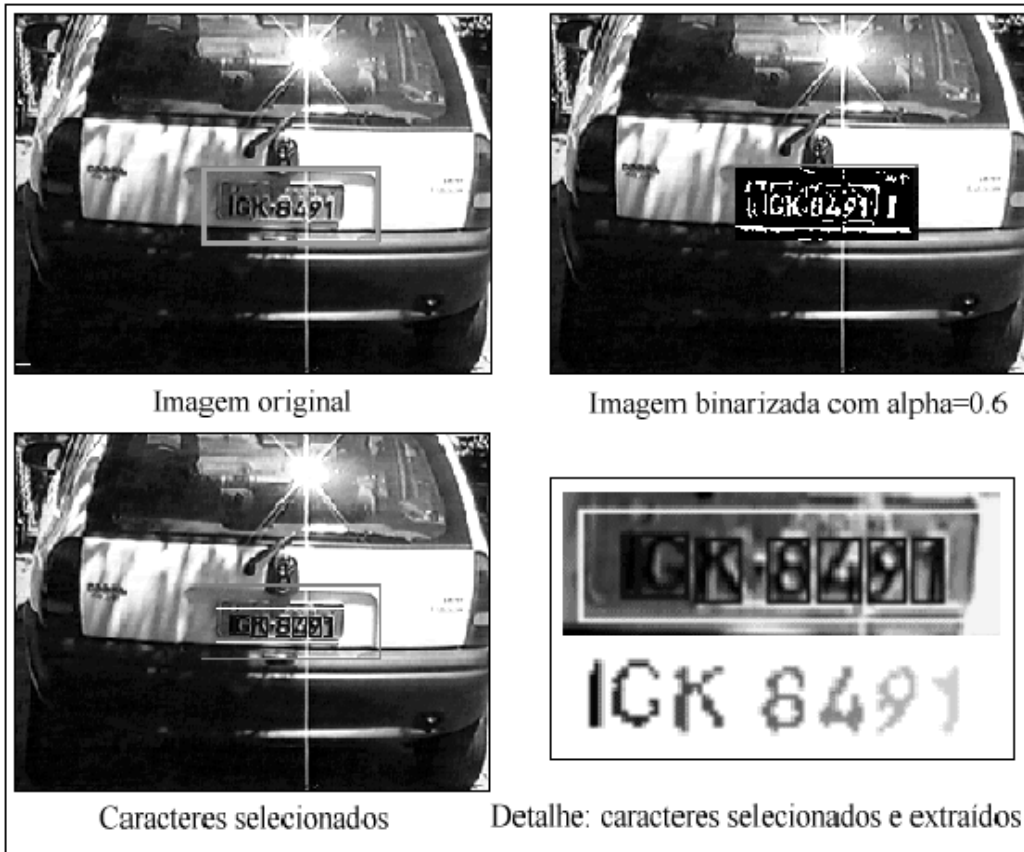


Fig. 7.10 – Etapas do sistema

Na figura 7.10 o algoritmo de localização por análise tonal obteve sucesso. Neste caso, o algoritmo de análise por dígitos se restringe apenas ao local indicado localizando corretamente os dígitos. O tempo de processamento para esta imagem fica reduzido a 6.6 segundos e a resposta da rede neural é IEK6491 (a resposta correta é IGK6491).

O algoritmo de localização da placa através da procura por dígitos possui um tempo de processamento variável entre 1 e 130 segundos e é dependente da complexidade da imagem analisada. Este algoritmo apresentou um desempenho de 97.7 % de sucesso na localização da placa de licença e 91.3 % de sucesso na extração correta dos 7 caracteres nela contidos.

7.3 REDES NEURAI

Os resultados dos dois tipos de redes treinadas, para um conjunto de 200 imagens, são apresentados no gráfico 1. O primeiro par de barras refere-se à rede que foi treinada sem as duas linhas de descrição dos caracteres. O segundo par de barras refere-se à rede que foi treinada com a descrição proposta no capítulo 4. A primeira barra, em cada par de barras, refere-se ao número de caracteres corretamente reconhecidos considerando a totalidade dos caracteres analisados (1400 caracteres). E a segunda barra, em cada par de barras, refere-se ao número de placas corretamente reconhecidas (sete caracteres corretamente identificados).

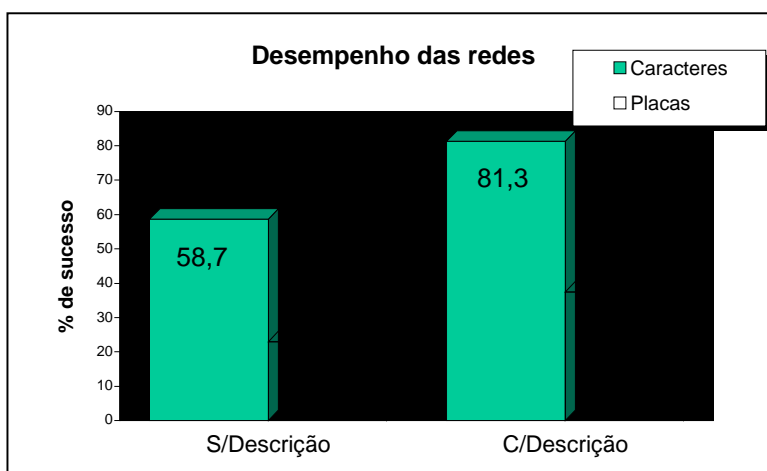


Gráfico 1 – Resultados comparativos das redes neurais

A rede neural utilizada atualmente obteve 37,5 % de sucesso na identificação correta de todos os sete caracteres contidos na placa de licença, como é o caso da figura 7.11, e 81,3 % de sucesso na identificação dos caracteres considerando a totalidade dos caracteres analisados (Fig. 7.12).



Fig. 7.11 – Identificação correta da placa



Fig. 7.12 – Identificação correta de seis caracteres da placa

7.4 RESULTADOS GERAIS E COMPARATIVOS

O sistema desenvolvido (SIAV) foi testado com um conjunto de 500 imagens e comparado com um *software* demonstrativo - Seecar - disponível no site <http://www.htsol.com> da empresa israelense High Tech Solutions. Este *software* foi testado com o mesmo banco imagens e os resultados são apresentados no gráfico abaixo.

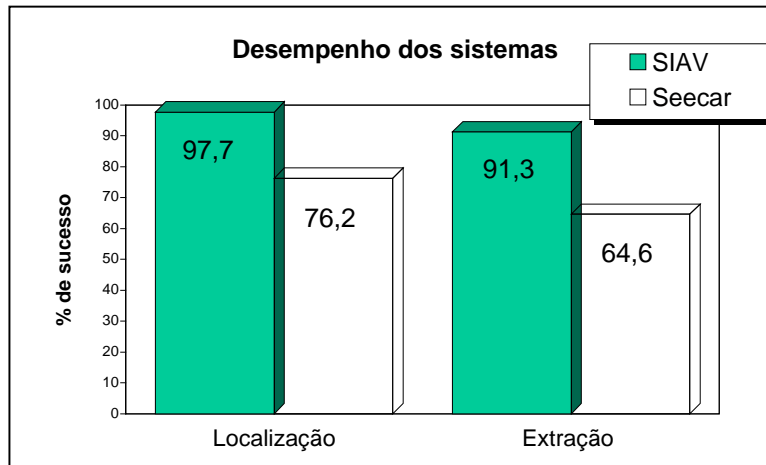


Gráfico 2 – Resultados comparativos dos sistemas

No primeiro par de barras têm-se as taxas de sucesso na localização correta do local da placa de licença, e no segundo par, temos as taxas de sucesso na extração correta dos 7 caracteres encontrados.

O tempo de processamento gasto pelo *software* Seecar para cada imagem é de aproximadamente 0.01 segundos. Esse desempenho é comparável ao de um sistema que se baseie apenas em uma análise por variação tonal para localizar e extrair os caracteres da placa.

CAPÍTULO 8

CONCLUSÃO E PERSPECTIVAS FUTURAS

8.0 CONCLUSÃO

O presente trabalho apresentou um conjunto de técnicas de processamento, análise e interpretação de imagem objetivando o desenvolvimento de um sistema de localização e leitura de caracteres alfanuméricos em imagens de veículos. Através do estudo de diversas técnicas existentes na literatura pôde-se apresentar aquelas que obtiveram melhores resultados e desenvolver um algoritmo próprio, baseado nas técnicas estudadas.

O algoritmo de localização da placa de licença através da análise da variação tonal em uma imagem mostrou-se muito rápido, na ordem de 0.03 segundos por imagem. Porém, este algoritmo não obteve a mesma eficiência que o algoritmo de análise por dígitos. Este, por sua vez, é extremamente lento comparado ao algoritmo anterior, além de não ser muito eficiente em imagens onde a distribuição de luz sobre a superfície da placa é heterogênea.

A solução adotada em relação ao problema de tempo de processamento foi a utilização do algoritmo de análise por variação tonal apenas para restringir o local analisado pelo algoritmo de análise de dígitos. Em aproximadamente 83 % dos casos a binarização e o processo de procura por dígitos é feito em uma imagem de 120x50 *pixels*, ao invés dos 320x240 *pixels* da imagem original. Quanto ao problema da distribuição de luz sobre a placa, a solução criada foi a utilização de um limiar α no algoritmo de binarização adaptativa, que é variado em função do número de caracteres encontrados. Quando este número não está de acordo com o número de caracteres esperados, o algoritmo incrementa o valor de α até encontrar um número de caracteres válido, ou até o limiar α exceder um limite. Este procedimento conseguiu, na maioria das vezes, “descolar” caracteres que, por ventura, estivessem “colados” a outros ou às bordas da placa, conferindo uma robustez para o sistema não encontrada nos algoritmos testados.

Como é evidente pela observação dos resultados apresentados, neste trabalho optou-se enfatizar o estudo das técnicas de processamento e análise de imagem em detrimento da técnica de inteligência artificial adotada. Deste modo, a solução apresentada através das redes neurais utilizadas serviu apenas como um complemento ao trabalho, possibilitando a visualização do sistema como um todo, localização da placa – extração dos caracteres – leitura dos caracteres. É claro que um estudo mais aprofundado deverá ser feito para a real utilização de um sistema deste tipo.

Os resultado apresentados pelo sistema são animadores. Em relação ao que tange o processamento de imagem propriamente dito, o objetivo foi plenamente alcançado. Apesar das dimensões das placas analisadas serem menores que as do sistema utilizado para comparação, 120x40 *pixels* no melhor caso, contra 190x56 *pixels*, o sistema possui uma taxa de sucesso na localização das placas de 97,7 %, superior aos 90 % relatado em [20]. A taxa de sucesso na extração dos caracteres contidos na placa é de 91.3 % muito próximo dos 92.5 % apresentados em [20].

Em função do algoritmo desenvolvido possuir um limiar α variável, o tempo de processamento de cada imagem varia de acordo com a complexidade da mesma. O tempo de processamento máximo apresentado (130 segundos) está superestimado. Nos testes feitos a variação da dimensão vertical dos caracteres da placa era de 21 *pixels*. Em uma situação apropriada de uso (câmera a uma distância fixa dos veículos), esta variação pode ser reduzida, minimizando o tempo gasto na rotina de localização dos dígitos.

8.1 PERSPECTIVAS DE TRABALHOS FUTUROS

O trabalho apresentado constitui um estudo dentro do contexto de identificação automática de veículos através da localização e leitura da placa de licença do mesmo. Como tal, pretende explorar os diferentes caminhos de implementação de um sistema desta natureza, sem preocupar-se em demasia com a performance apresentada. Por este motivo, diversas modificações podem ser sugeridas como trabalho futuro.

Antes das sugestões relativas as etapas do sistema apresentado, a utilização de imagens onde as dimensões da placa de licença são maiores que as dimensões utilizadas atualmente, deve melhorar a resolução dos caracteres, permitindo uma melhor extração dos mesmos e conseqüentemente um aumento na taxa de acertos de leitura.

Na etapa de localização poderia-se gerar uma imagem negativa da imagem binarizada para permitir que o sistema procure placas de fundo escuro com caracteres claros. A etapa de localização dos dígitos seria feita duas vezes, uma para cada imagem aumentando o tempo de processamento, porém, permitiria que o sistema trabalhasse com qualquer tipo de placa. Além disso, poderia-se analisar a tonalidade dos dígitos encontrados antes de fechar um grupo válido de caracteres. Aqueles dígitos que possuíssem um tom de cinza semelhante seriam agrupados, ou seja, o algoritmo de análise dos dígitos encontrados levaria em conta além das características dimensionais e posições relativas, o tom do dígito. Este procedimento evitaria, na maioria dos casos, que o sistema se confundisse com possíveis bordas laterais das placas, o que ocorre atualmente causando um engano em relação aos caracteres da placa.

Na etapa de extração dos caracteres poderia-se extraí-los da imagem binarizada sem redimensioná-los, e descrevê-los para a rede neural através de suas formas. Na literatura pesquisada há diversas propostas para isso, porém uma em particular parece interessante. A descrição dos caracteres através de coeficientes de Fourier [6] tem como vantagens as propriedades da transformada, como invariância a rotação, escalonamento, deslocamento, etc.

E, finalmente, na etapa de leitura dos caracteres sugere-se um estudo mais aprofundado nas redes utilizadas para reconhecimento de caracteres alfanuméricos. Este é um assunto muito interessante e pode complementar o sistema aproveitando a elevada taxa de possibilidade de reconhecimento de caracteres (extração correta dos sete caracteres).

APÊNDICE A – SOFTWARE

O sistema apresentado neste trabalho foi desenvolvido em linguagem C padrão em uma versão para DOS e em C++ para Windows. Esta última apresenta as principais etapas de localização, extração e reconhecimento de uma forma fácil e atraente, permitindo ao usuário alterar variáveis como o tamanho dos caracteres que estão sendo procurados, o valor de θ da rotina de análise por variação tonal, e α inicial, final e incremento.

O *software* permite que o usuário adquira imagens de uma câmera ou vídeo cassete conectado à placa de aquisição SE100 da Creative. Além disso, o *software* gera um relatório final com as etapas executadas e seus respectivos tempos de processamento. A figura abaixo mostra a interface do *software* desenvolvido.

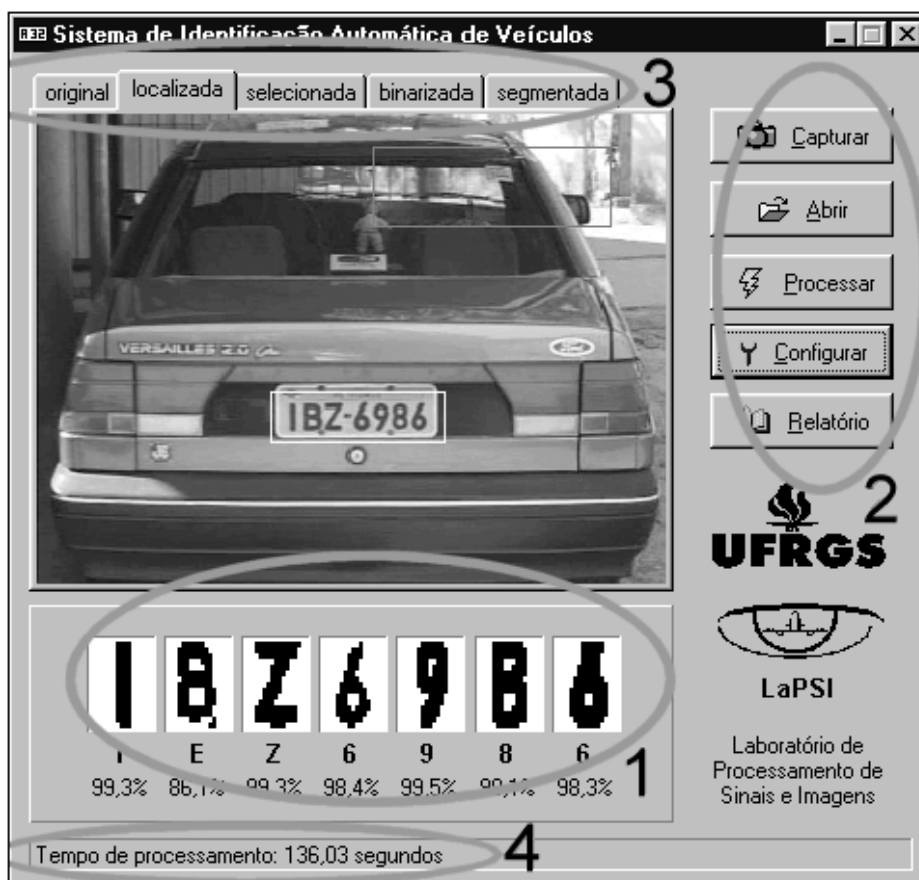


Fig. A1 – Interface do *software*

1 – Mostrador dos caracteres extraídos e redimensionados, e abaixo o resultado do reconhecimento das rede neurais.

2- Opções de operação:

Capturar uma imagem proveniente de algum dispositivo acoplado a placa de aquisição.

Abrir alguma imagem em formato bitmap (.BMP) disponível em disco (24 ou 8 bits).

Processar a imagem presente na tela principal do *software* a fim de localizar e reconhecer o conteúdo da placa do veículo.

Configurar os valores de θ e α (gama no caso da figura abaixo) e a dimensão vertical mínima e máxima dos caracteres procurados.

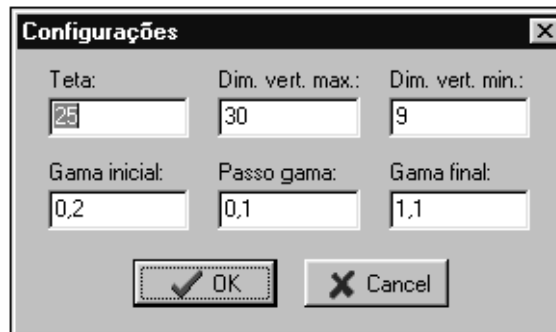


Fig. A2 – Janela de configuração

Relatório das etapas executadas com os respectivos tempos de processamento.

3- Menu de imagens com as principais etapas de localização e realce da placa, seleção dos caracteres e extração dos mesmos.

4- Barra informativa do processamento executado em tempo real. Nesta barra o usuário acompanha cada passo que está sendo executado entendendo melhor o comportamento do sistema.

APÊNDICE B - EQUIPAMENTOS

- Câmera de vídeo: marca gradiente, modelo CGP 9602, CCD, gravação VHS-C.
- Câmera fotográfica digital: marca Dycam, modelo 10-C.
- Placa digitalizadora de vídeo (*frame grabber*): (1) marca Zoltrix, modelo Face to Face, resolução 320x240 *pixels*, RGB 24 bits (2) marca Creative, modelo SE 100 VideoBlaster, resolução 320x240 *pixels*, RGB 24 bits.
- Microcomputador: AMD K6-2 400 MHz, com 64 MB de RAM, disco rígido de 3.5 Gbytes.

BIBLIOGRAFIA

- [1] MESSELODI S., MODENA M., **Context Driven Segmentation in Text Recognition**, Istituto per la Ricerca Scientifica e Tecnologica, ITALY, 21/08/1996, <http://hera.itc.it:3003/~messelod/OCR/TextRecognition.html>
- [2] TURNER S. M., **Advanced Techniques for Travel Time Data Collection**, Transportation Research Record 1551, p. 51-58.
- [3] SEECAR SYSTEM , High Tech Solutions - <http://www.htsol.com>
- [4] LIM S. J., **Two-Dimensional Signal and Image Processing**, Prentice Hall PTR.
- [5] CASTLEMAN K. R., **Digital Image Processing**, Prentice Hall, 1996.
- [6] GONZALEZ R. C., WOODS R. E., **Digital Image Processing**, Addison-Wesley Publishing Company, Chapter 3, September 1993.
- [7] SHULDINER P. W., D'AGOSTINO S. A., WOODSON J. B., **Determining Detailed Origin-Destination and Travel Time Patterns Using Video and Machine Vision License Plate Matching**, Transportation Research Record 1551, p. 8-17.
- [8] NIBLACK W., **An Introduction to Digital Image Processing**, Prentice Hall, 1986, p.155-116
- [9] COMELLI, P.; FERRAGINA, P.; et al, **Optical Recognition of motor vehicle license plates**, IEEE Transactions on Vehicular Technology, vol. 44, No. 4, p. 790-799, November 1995.
- [10] LIU Y., SRIHARI S. N., **Document Image Binarization Based on Textures Features**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 5, May 1997
- [11] TRIER O. D., JAIN A. K., **Goal-Directed Evaluation of Binarization Methods**, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, No. 12, December 1995, p. 1191-1201

- [12] KERTÉSZ A. , KERTÉSZ V., MULLER T., **An On-Line Image Processing System For Registration Number Identification**, Proceedings of the 1994 IEEE International Conference on Neural Networks, Orlando,FL,USA, Part 6 (of 7) p. 4145-4148.
- [13] HAYKIN S., **Neural Networks A comprehensive foundation**, Macmillian College Publishing Company, Inc, 1994
- [14] HEIKKONEN J., BULSARI A., Editorial **Special Issue on Neural Networks for Computer Vision Applications**, Pattern Recognition Letters 17 (1996) 317-318
- [15] PLAUT D.C, NOWLAN S.J., HINTON G.E., **Experiments on Learning by Back propagation**, Computer Science Department Carnegie-Mellon University Pittsburgh, PA 15213, June 1986, Technical Report CMU-CS-86-126
- [16] FATHY M., SIYAL M.Y., **Measuring traffic movements at junctions using imagem processing techniques**, Pattern Recognition Letters 18 (1997) 493-500
- [17] BARROSO J., RAFAEL A., DAGLESS E. L., BULAS-CRUZ J., **Number plate reading using computer vision**, <http://www.utad.pt/~jbarroso/html/isie97.html>, 7/07/1999.
- [18] DRAGHICI S., **A Neural Network based artificial Vision System for License Plate Recognition**, International Journal of Neural Systems, Vol. 8, No. 1 (February, 1997) p. 113-126.
- [19] SOH Y. S., CHUN B. T., YOON H. S., **Design of Real Time Vehicle Identification System**, Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics, San Antonio,TX,USA, Part 3 (of 3), p. 2147-2152
- [20] COETZEE C., BOTHA C., WEBER D., **PC Based Number Plate Recognition System**, Proceedings of IEEE International Conference on Industrial Electronics 1998.
- [21] TRIER O. D., TAXT T., **Evaluation of Binarization Methods for Document Images**, Transactions on Pattern Analysys and Machine Intelligence, vol. 17, No. 3, March 1995, p. 312-315