

Reconhecimento Automático de Placas Veiculares

Luismar Sebastião Ferreira *

* UniverCidade (Centro Universitário da Cidade do Rio de Janeiro)
Núcleo de Projetos e Pesquisa em Aplicações Computacionais

e-mail: tecluismar@hotmail.com

Luciano Pereira Soares* ‡

‡ Tecgraf, DI - Departamento de Informática
PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro)

e-mail: lpsoares@tecgraf.puc-rio.br

Abstract

A set of techniques and algorithms are described for an automatic recognition of characters in license plates. Developed resources such as tracking polygons in images using the characteristic color of the plates with neighborhood pixel routines have presented efficient results. Procedures to accurately locate and recognize the characters are also presented. The research was based on neural networks combined with techniques of digital image processing by automated agents that expose the results immediately for the end users.

Resumo

Um conjunto de técnicas e algoritmos são apresentados para o reconhecimento automático de caracteres em placas de veículos. Foram desenvolvidos recursos de rastreamento de polígonos em imagens através da cor característica das placas. Tais recursos se mostraram muito eficientes, usando para isso rotinas de vizinhança de pixels adaptadas. Procedimentos para localizar e reconhecer com precisão os caracteres também são apresentados. A pesquisa baseou-se em redes neurais, aliadas às técnicas de processamento de imagens por agentes que permitem expor os resultados imediatamente aos usuários.

1. Introdução

O presente projeto de pesquisa teve como finalidade desenvolver um sistema inteligente capaz de reconhecer automaticamente os símbolos alfanuméricos a partir de imagens digitalizadas de veículos. O sistema, ao receber uma imagem digitalizada contendo um veículo, é capaz de identificar e separar a placa do veículo do restante da cena, normalizando e enquadrando a imagem da placa, para então reconhecer a sequência de símbolos alfanuméricos que identificam o veículo. O sistema gera uma saída com a informação referente à identificação da placa, podendo automatizar outros processos em cadeia.

Técnicas de inteligência artificial [1], como agentes para localizar e segmentar as imagens e redes neurais [2], são usadas para reconhecimento ótico dos caracteres.

O reconhecimento automático das placas de veículos tem inúmeros usos, como por exemplo: calcular a

permanência de veículos em um estacionamento, automatizar a entrada e saída da frota de uma empresa em seu estacionamento, controlar a entrada de veículos autorizados em algum um local, identificar veículos furtados ou irregulares transitando em via pública.

O capítulo a seguir apresenta o sistema de placas brasileiro, e no capítulo 3 são listados alguns trabalhos relacionados. No capítulo 4 serão apresentados os conceitos dos algoritmos utilizados na pesquisa e no 5 como eles foram desenvolvidos. No capítulo 6 são apresentados os resultados e no 7 uma breve conclusão da pesquisa.

2. Padronizações de placa veicular

Segundo o CONTRAN [3] todos os veículos fabricados ou emplacados a partir de 01/08/2007 no Brasil devem seguir a resolução 231 de 15/03/2007. Nesta resolução foram estabelecidos padrões para as dimensões das placas de motos e carros. Também ficou estabelecido o modelo para as letras e números a serem usados, assim como seus tamanhos e posicionamentos sobre o retângulo da placa. A mesma resolução também estabelece as cores para o fundo das placas e para as letras de acordo com a categoria de cada veículo, além da obrigatoriedade de película retro-refletiva sobre as mesmas.

2.1. Codificação da placa

O código da placa é o resultado das combinações de 3 letras (26 símbolos, de A-Z) e 4 números (10 símbolos, de 0-9), sendo que cada combinação é distribuída pelo CONTRAN [3] para garantir o padrão de registros dentro da federação. Os diversos elementos da placa são padronizados conforme mostra a Figura 1. Estes elementos são impressos fisicamente em uma chapa de metal, provocando o efeito de alto relevo, e pintados.

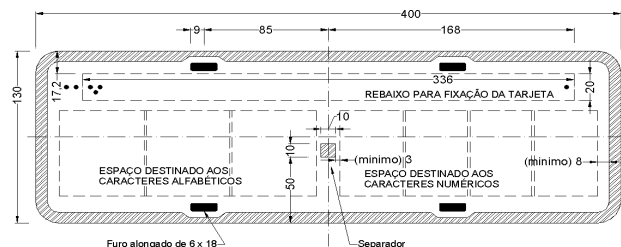


Figura 1 – Dimensionamento da placa de carros (mm).

3. Trabalhos relacionados

TRENTINI et al. [4] desenvolveram um sistema de visão computacional para reconhecimento automático de placas veiculares em ambiente externo. Filtragens em um pré-processamento no domínio de frequência permitiu suavizar e eliminar eventuais ruídos da imagem, melhorando a localização e o reconhecimento das placas. Com isso são eliminadas também saliências e reentrâncias que atrapalham o algoritmo de detecção de bordas.

Para o rastreamento são realizadas buscas espaciais comparando as dimensões e atributos dos elementos pertinentes nas imagens com as proporções das dimensões e padrões de uma placa. A correção da perspectiva da placa apresenta um inconveniente que provoca pequenos “buracos” na imagem final.

Os caracteres são reconhecidos através do algoritmo de Random Trees, que é um classificador com base em árvore de decisão. O resultado foi positivo em aproximadamente 92% dos veículos testados.

POLIDORIO e BORGES [5] fazem a aquisição de uma imagem binarizada, fazendo uma análise de contraste que foca no centro inferior da imagem para localizar a placa do veículo. Após a separação dos caracteres o esqueleto do caractere é procurado para ser reconhecido.

O reconhecimento é feito por análise sintática, se processando apenas um conjunto representativo das primitivas. Cada objeto encontrado tem seus pontos de interesses identificados, sendo independentes de rotação, translação e escala. Porém o processo é ineficiente para os caracteres {o, 0}, {W, M}, {Z, 7, J}, {6, 9} quando analisados, não levando em consideração itens de rotação e posicionamento na cadeia de combinações. Os autores apontam um resultado de aproximadamente 95% de eficácia no reconhecimento das placas.

CONCI e MONTEIRO [6] usaram várias técnicas de reconhecimento, que vão desde o reconhecimento da posição da placa até as métricas estabelecidas para definir o padrão que mais se aproxima de cada caractere da imagem a ser reconhecida.

O correto ponto de corte da binarização da imagem é calibrado por uma ferramenta de histograma, junto a uma técnica de erosão aplicada ao processamento. Para o reconhecimento do símbolo é feito o cálculo do momento de área de cada objeto, logo então este é comparado com o momento já conhecido para cada símbolo, escolhendo-se o símbolo mais próximo. Com o conjunto destas técnicas foi obtido sucesso de 100% no reconhecimento das placas testadas.

GUINGO et al. [7] usaram imagens de radares fotográficos da Companhia de Engenharia de Tráfego do Município do Rio de Janeiro (CET-RIO). Como o trabalho foi feito sob condições reais, fatos como neblina, chuva, luminosidade, ângulo, distância e velocidades do instante da captura foram avaliadas.

A localização da placa, na imagem adquirida, é feita manualmente recortando-se a região da placa e salvando em um novo arquivo. A segmentação dos caracteres da placa é feita por uma verificação na frequência de pixel com preenchimento acumulado por colunas e detectando pontos de corte. Assim é gerando um polígono retangular delimitando cada caractere.

Para a extração de características para cada caractere foi usado um algoritmo que deriva do algoritmo de detecção de contorno que, por sua vez, verifica a distância do contorno do objeto até o encontro com a borda do polígono previamente projetado.

Como o caractere pode encontrar-se rotacionado na imagem é feita a extração de características em 4 ângulos diferentes: 0°, 45° 90° e 135°, partindo da borda de mira do polígono para o centro até atingir o contorno do símbolo ou objeto. A taxa de acerto para letras foi de 88.9% e de número foi de 96.13%. A rede para as letras tem dificuldade de distinguir os caracteres B, D, O, G, I, L, J, T, Y, V, e para os números a dificuldade centrou-se na distinção entre 8 e 0.

4. Algoritmos de reconhecimento

O sistema desenvolvido é automatizado por multi-agentes, com um agente coordenador, que identifica novos cenários disponíveis no sistema de imagens, disparando um conjunto de ações, desde o tratamento da imagem original até a apresentação do código da placa.

O agente coordenador repete infinitamente buscas a cada 0,1 segundo, visto que esse é o tempo máximo que o processo completo leva para verificar uma cena.

Quando o agente coordenador percebe uma nova cena, este passa uma ordem para o agente localizador de placas, que localiza e segmenta a parte que corresponde a placa do veículo e responde ao coordenador sobre o término e situação da busca. O agente coordenador usa a informação recebida do localizador para decidir se solicita ou não a identificação do código da placa pelo agente identificador. O agente coordenador precisa ser criado antes dos outros agentes, assim um temporizador força que o coordenador seja criado antes dos outros operários. O sistema multi-agentes foi desenvolvido em Jade[8].

Os agentes finalizam sua atuação ao encontrar e apresentar o código da placa ou até esgotar, sem sucesso, todas as possibilidades de busca pelo código. A seguir são apresentadas as principais etapas do sistema:

4.1. Busca pela placa

Diversas propostas e ensaios para localizar a placa foram efetuados durante a pesquisa. A busca pela proporcionalidade dos objetos dos veículos e a extração dos polígonos da cena foram desenvolvidas e testadas, como se pode ver na Figura 2. Contudo, o melhor resultado foi aquele baseado na cor de fundo das placas.

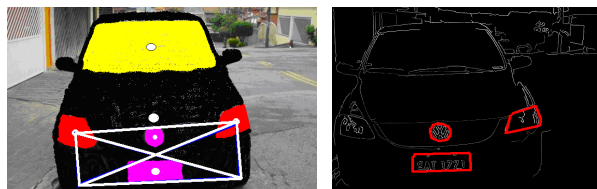


Figura 2 – Busca por proporções e poligonização.

Uma estrutura de dados em 3 dimensões armazena informações referentes aos resultados dos processamentos necessários para encontrar a placa contida na imagem. Esta estrutura é composta por uma matriz bidimensional utilizada para armazenar resultados temporários e todos os cálculos intermediários de rastreamento.

A cor da placa é procurada levando em consideração a sua conectividade com os vizinhos similares [9], bem como os padrões de cores das placas veiculares. Para isso cada pixel terá suas componentes de cores avaliadas, e se estiver atendendo ao critério de busca, o pixel em questão é marcado como uma região candidata. Os valores das cores desejadas são ajustados dinamicamente de acordo com o aprendizado do sistema durante sua execução.

Foi desenvolvido um algoritmo que percorre todos os pixels da imagem original analisando se a tonalidade tem possibilidade de ser uma placa de cor cinza, marcando assim esta informação na célula que corresponde ao pixel na estrutura de dados. Após isso o algoritmo verifica e marca as células candidatas que são conexas com outras candidatas a serem parte da placa.

Posteriormente as células são percorridas novamente e agrupadas por uma identificação de grupo conexo, formando um conjunto de elementos com as mesmas características. Cores distintas para cada grupo identificado são apresentados na Figura 3.



Figura 3 – grupos candidatos coloridos.

Como o resultado ainda não foi o desejado, então é feita a média aritmética dos pixels vizinhos. Um algoritmo faz a ordenação das células para posteriormente encontrar as coordenadas dos quatro cantos e assim calcular a razão das dimensões de cada polígono, conforme apresentando nos textos em azul na Figura 4.



Figura 4 – Polígono sobre cada grupo conexo.

Como as imagens são escaladas para uma resolução fixa, os polígonos com quantidade de pixels insuficiente para identificação de placas são descartados, assim reduzindo a quantidade de polígonos candidatos. A regra usada foi que um polígono com menos de 100 pixels na horizontal não tem condições de ser placa.

Após a seleção de todos os grupos conexos de pixels candidatos, cada grupo candidato terá uma identificação. A Figura 5 apresenta um resultado para placas cinzas.



Figura 5 – Original e grupos cinzas de pixels.

Após a identificação, os pixels externos formarão um polígono candidato como ilustrado na Figura 6.



Figura 6 – Primeiro, segundo e terceiro polígono.

Caso não exista polígono candidato para a placa de cor cinza, as operações são repetidas para vermelho, branco, preto e azul, sendo que o filtro aplicado para busca do pixel candidato será específico para cada cor de fundo.

Todos os polígonos que não possuem proporção de um retângulo de placa são descartados. Para isso é feito um cálculo de dimensões para verificar se o polígono tem aproximadamente as proporções de uma placa (400x130mm), ou seja, a razão da largura sobre a altura é dada por $3,07 \pm 10\%$ de tolerância. Estes polígonos são marcados como forte candidatos a placa.

4.2. Nivelar e corrigir o ponto de fuga

O agente dispara um algoritmo para reconhecimento dos 4 cantos do polígono candidato, checando quais são agudos e próximos de 90 graus, tendo assim condições de aplicar outro algoritmo para correção de nivelamento, escala e ponto de fuga. Também é feita a correção da rotação e o descarte do contorno desprezível. Após isso a imagem é escalada para 400 por 130 pixels e uma máscara com um pixel para cada mm² referente na placa é aplicada. A Figura 7 ilustra o que é esperado.

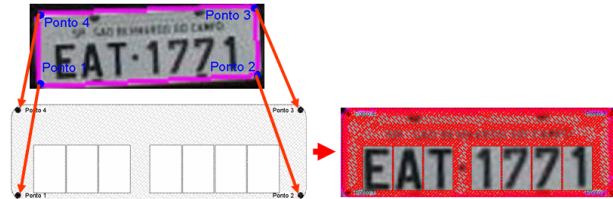


Figura 7 – Nivelamento pelos pontos de controle.

4.3. Extrair da imagem letras e números

Para validar os polígonos candidatos a imagem é binarizada se convertendo para preto e branco, fazendo assim surgir espaços vazios entre os elementos da placa que permitem localizar os caracteres e elementos da mesma. Foi levado em consideração que, se um polígono candidato não tem mais que 10 grupos de pixels (3 letras + 4 números + fundo da placa + letras da cidade + letras estado), então ele não pode ser uma placa.

O algoritmo de segmentação na horizontal faz uma busca de cima para baixo e delimita a linha com a menor quantidade de pixels pretos entre a linha 35 e 55. O agente fará a soma e a distribuição dos pixels de cor preta nas linhas. Isso visa identificar se existe uma linha de corte que coincida com a linha de corte horizontal da máscara. Este corte separa o nome do Município e UF do restante da placa, conforme demonstrado pela Figura 8.

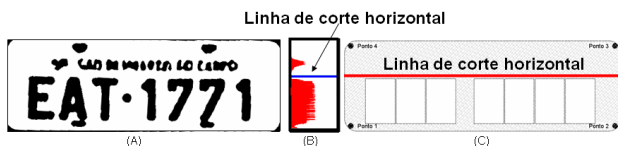


Figura 8 – (A) imagem binarizada, (B) distribuição dos pixels da horizontal, (C) linha de corte horizontal.

Em um próximo passo, o agente faz a soma e a distribuição dos pixels pretos nas colunas da parte inferior da linha de corte horizontal. Isso visa identificar os seis pontos de corte, e se eles são próximos aos pontos de corte verticais da máscara, conforme Figura 9. O algoritmo faz 3 buscas sequenciais relativas ao último ponto de corte. Este algoritmo pula 49 pixels e considera como ponto de corte a coluna com a menor quantidade de pixels pretos de 42 a 56 na largura, relativamente ao ponto de corte.

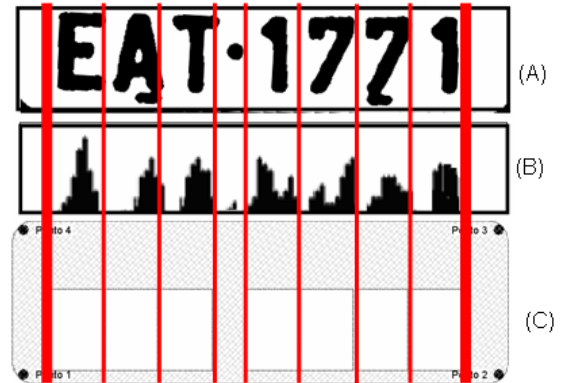


Figura 9 – (A) imagem binarizada, (B) distribuição dos pixels da vertical, (C) linhas de corte na vertical.

O agente é capaz de verificar se é necessário algum ajuste de translação na horizontal e vertical para melhorar o posicionamento em relação à máscara, analisando a sobreposição da distribuição sobre a máscara de corte.

Uma vez a imagem bem posicionada e com seus símbolos não sobrepostos sobre a máscara, outro algoritmo translada todos os símbolos para o extremo superior esquerdo do espaço reservado (Figura 10). Esta operação visa padronizar o posicionamento inicial dos caracteres para reconhecimento pelas redes neurais.

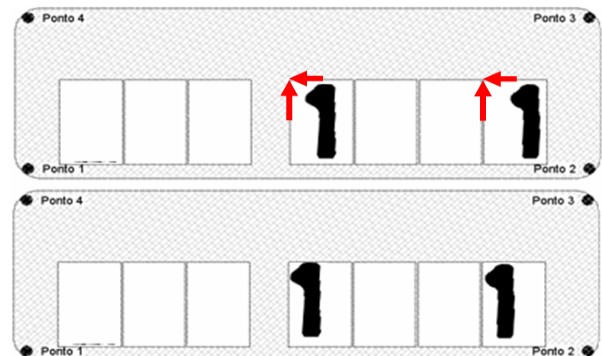


Figura 10 – Deslocando símbolos em relação à máscara.

4.4. Reconhecimento dos algarismos

Como os espaços destinados às letras são padronizados, a matriz que representa a imagem da letra tem 65x58 pixels. Esses pixels serão analisados pela rede neural artificial treinada para reconhecer letras no canto superior esquerdo. Esta rede terá em sua camada de saída 26 células. No final, uma única célula ativa identificará a letra. A rede apresenta o melhor comportamento seguindo o modelo *perceptron* de 3 camadas [10]. O treinamento é realizado com *backpropagation*. O mesmo acontece com os números de 65x44 pixels, com saída em 10 células.

5. Implementação da rede neural artificial

As redes foram implementadas através de cinco procedimentos com ferramentas distintas (Figura 11).

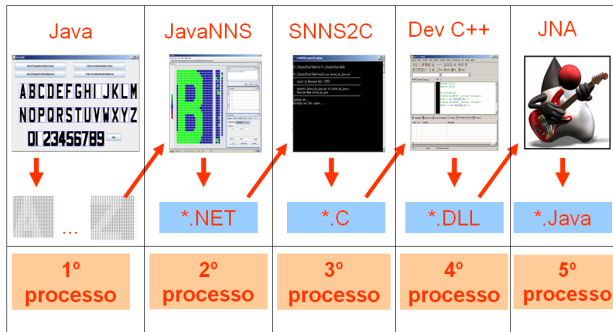


Figura 11 – Processos para implementação da RNA.

Para treinar a rede foi necessário criar os arquivos com os parâmetros de entrada, que contém amostras dos objetos que a rede usa para fazer sua aprendizagem. Diversas amostras de letras e números foram geradas, variando o estilo do DETRAN “mandatory” com recursos como erosão e dilatação. Uma entrada exemplo é apresentada na Figura 12.

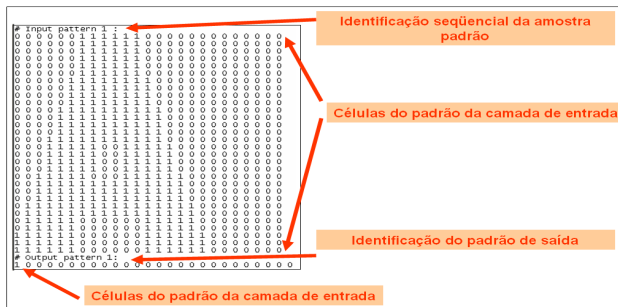


Figura 12 – Padrão de entrada para a letra “A”.

A rede neural pode ser configurada, treinada e testada com o JavaNNS [11] (Java Neural Network Simulator), resultando em um arquivo (.net) com as características da rede e os pesos sinápticos. A conversão deste arquivo em algoritmo foi realizada através do programa SNNS2 [12], que gera o código em linguagem C. Para usar o código C em Java, este foi compilado para uma biblioteca de vínculo dinâmico DLL (*dynamic-link library*), reintegrada no Java por JNA (Java Native Access).

Originalmente trabalhou-se com 3770 células (58x64), porém o sistema ficou excessivamente lento. Assim se reduziu a camada de entrada para a rede das letras a 750 células, dispostas 25 na horizontal e 30 na vertical.

Para a camada intermediária foram criadas 90 células e a camada de saída foi criada com 26 células. As conexões da rede seguiram o padrão *feed-forward* ou cíclico, sendo este o modelo proposto para uma rede *perceptron*. A

Figura 13 exibe a criação de uma rede em proporções minimizadas.

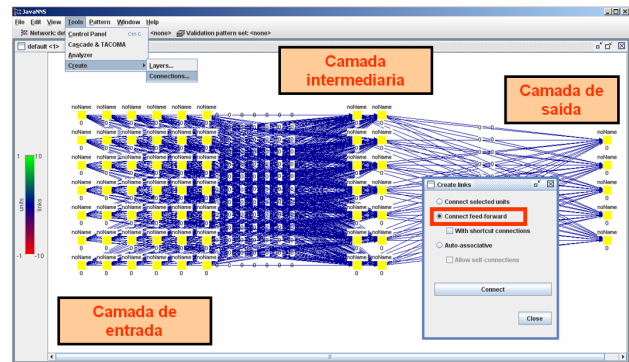


Figura 13 – Criando as conexões feed-forward.

As interfaces do JavaNNS (Figura 14), são de grande utilidade para fazer a análise do andamento do treinamento e também da configuração da rede.

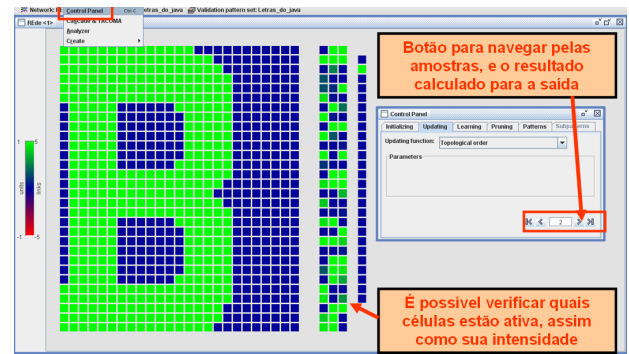


Figura 14 – Análise das células das camadas.

A rede produzida aprendeu com 930 épocas e levou 5 minutos, e sua evolução é exibida na Figura 15. Podemos observar que a maioria das células da camada intermediária está atuando no processo.

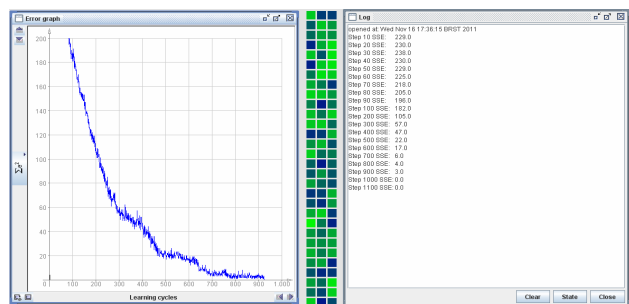


Figura 15 – Gráfico do sucesso no treinamento.

O objetivo foi alcançado por usar um valor pequeno na variação inicial dos pesos das conexões, e também por usar um peso pequeno para fazer a retro-distribuição dos erros encontrados em cada época.

6. Resultados

Inúmeros testes foram conduzidos, verificando-se a eficiência do processamento de imagens, o treinamento das redes neurais e também a criação dos agentes.

O sistema foi testado com placas de veículos em imagens adquiridas de 3 e 7 metros de distância, providas de iluminação natural. A câmera foi posicionada entre 1,5 e 2,5 metros de altura em relação ao piso do veículo. Os veículos estavam em rotação de até 15 graus. A Figura 16 mostra dois exemplos de resultados.



Figura 16 – Dois testes em carros.

Todas as letras e números que serviram de teste obtiveram sucesso de 100% no reconhecimento, contudo somente testes com placas sem artefatos foram realizados. Uma identificação em PC convencional leva cerca de 0,25s para as letras, e mais 0,16s para os números.

7. Conclusões

A pesquisa foi orientada para um sistema capaz de efetuar o reconhecimento automático de placas de veículos. Diversas técnicas de processamento de imagens e inteligência artificial foram adaptadas e compostas.

Dentre as principais contribuições podemos citar os sistemas multi-agentes que possuem a tarefa de localizar a placa sobre a imagens adquirida, segmentar a placa na imagem, localizar as letras e números, recortar os símbolos e efetuar seu reconhecimento por redes neurais.

A implementação dos algoritmos mostrou-se eficiente no reconhecimento de letras e números nas placas, e também na localização da placa frente às imagens dos veículos testados. Após a geração de diversos caracteres para o aprendizado da rede, os testes mostraram uma eficiência de 100%, não se constatando erros no reconhecimento de caracteres para imagens com resolução e qualidade aceitável.

Como trabalho futuro pretende-se utilizar placas amassadas, condições de neblina e chuva, além de iluminação diferenciada para testar a robustez dos algoritmos.

Referências

- [1] G.F. Luger, "Intelligence Artificial – Estruturas e estratégia para a solução de problemas complexos", 4ª Ed. Bookmam, 2004
- [2] A.P. Braga, A.P. Carvalho e T.B. Ludermitr, "Redes neurais Artificiais: Teoria e Aplicações", LTC, 2000;
- [3] CONTRAN, Conselho Nacional de Trânsito - Resolução 231 de 15 de março de 2007
- [4] V.B. Trentini, L.A.T. Godoy e A.N. Marana, "Reconhecimento Automático de Placas de Veículos", VI Workshop de Visão Computacional. 2010
- [5] A.M. Polidório e D.L. Borges, "Reconhecimento de Caracteres Alfanuméricos para Identificação de Placas de Veículos", SIBGRAPI 1996, v. 1. p. 363-364.
- [6] A. Conci e L.H. Monteiro, "Reconhecimento de placas de veículos por imagem", CONEM 2004, agosto de 2004
- [7] B.C. Guingo, J Rodrigues e A.C.G. Thomé, "Técnicas de Segmentação de Imagens, Extração de Características e Reconhecimento de Caracteres de Placas de Veículos". VII Simpósio de Informática, Novembro, 2002. p. 91-96.
- [8] F. Bellifemine, A. Poggi e G. Rimassa, JADE: a FIPA2000 compliant agent development environment. V international conference on Autonomous agents (AGENTS '01). ACM, New York, USA, pp216-217. 2001
- [9] O. Marques Filho e H. Vieira Neto, "Processamento Digital de Imagens". Rio de Janeiro: Brasport, 1999.
- [10] F. Rosenblatt, "The Perceptron: A probabilistic model for information storage and organization in the brain". Psychological Review 65: 386-408, 1958
- [11] Wilhelm-Schickard-Institute for Computer Science (WSI), Java Neural Network Simulator, www-ra.informatik.uni-tuebingen.de/downloads/JavaNNS.
- [12] A. Zell, G. Mamier, M. Vogt, N. Mache, R. Hubner, K. U. Herrmann, T. Soye, M. Schmalzl, T. Sommer, A. Hatzigeorgiou, S. Döring, and D. Posselt. Snns: Stuttgart neural network simulator, Universität Stuttgart, 1994.