

RECONHECIMENTO DE PLACAS AUTOMOTIVAS UTILIZANDO REDES NEURAIAS ARTIFICIAIS

Rogério Bernardes Andrade¹; Cristiano Rodrigues Flores Maciel¹; Bruno Alves Romero²

^{1,2} Faculdade de Talentos Humanos - FACTHUS, Uberaba (MG), Brasil

rbandrade@facthus.edu.br, cristiano@maciel.eti.br, bruno.romero@facthus.edu.br

RESUMO: O presente trabalho descreve uma nova metodologia para desenvolvimento de sistema de identificação de placas de veículos automotores utilizando conceitos de processamento de imagens e inteligência artificial. Estes conceitos incluem pré-processamento, segmentação, extração de características e redes neurais artificiais. O principal objetivo deste trabalho é demonstrar o uso técnicas para reconhecimento de padrões com grande aplicabilidade. Os conceitos aqui apresentados são independentes de hardware, sistema operacional e linguagem de programação, o que facilita o desenvolvimento e operação e permite a utilização em diversos setores, sejam públicos, como fiscalizadores de trânsito, ou privados, como estacionamentos e outros.

PALAVRAS CHAVE: Processamento de imagens; Reconhecimento de padrões; Redes neurais artificiais.

RECOGNITION OF PLATES OF MOTOR VEHICLES USING ARTIFICIAL NEURAL NETWORKS

ABSTRACT: The present work describes a new methodology for development of automotive vehicle identification plate system using concepts of image processing and artificial intelligence. These concepts include preprocessing, segmentation, feature extraction and artificial neural networks. The main objective of this work is to demonstrate the techniques used for pattern recognition with wide applicability. The concepts presented here are independent of hardware, operating system and programming language, which facilitates the development and operation and allows use in various sectors, whether public, as traffic inspection, or private, such as parking lots and others.

KEYWORDS: Artificial neural networks; Image processing; Pattern recognition.

INTRODUÇÃO

Apesar de todo o avanço e popularização dos sistemas de informação, os veículos brasileiros ainda não possuem oficialmente um sistema de identificação digital. Mesmo com alguns projetos de identificação eletrônica, como o SINIAV (Sistema Nacional de Identificação de Veículos), a forma de fiscalização padrão utilizada é através o reconhecimento analógico das placas de identificação. O Decreto-Lei Nº 237 de 28 de fevereiro de 1967 determinou, entre outras, a criação do RENAVAM (Registro Nacional de Veículos Automotores) e a utilização de placas para a identificação e, mesmo com algumas mudanças, este modelo continua sendo utilizado até hoje.

Segundo informações do DENATRAN (Departamento Nacional de Transito) a frota nacional de veículos em Julho de 2014 era de aproximadamente 84 milhões, somando todos os tipos de veículos com placas de identificação. Os sistemas de controle de fiscalização, tráfego e furto de veículos em determinadas regiões brasileiras demandam altos custos com agentes, tanto públicos, quanto privados.

O objetivo deste trabalho é apresentar uma metodologia de desenvolvimento de um software capaz de reconhecer os padrões dos caracteres das placas de identificação veicular para auxiliar no controle e

fiscalização, permitindo consultar a situação do veículo em tempo real e sem a obrigatoriedade da intervenção humana.

MATERIAIS E MÉTODOS

O aplicativo para a demonstração deste artigo foi desenvolvido utilizando a plataforma e linguagem de programação Java versão 7 da Oracle. O desenvolvimento e os testes foram executados em um notebook ASUS com processador Intel Core i7 com 8 GB de memória. É importante mencionar que a ideia central do software é indiferente da linguagem de programação escolhida e, a fim de tornar a técnica portátil para qualquer outra linguagem e/ou plataforma, não foram utilizadas funções exclusivas da linguagem e nem bibliotecas de terceiros.

DESENVOLVIMENTO

Para o reconhecimento eletrônico das placas de identificação, foi criado um software de demonstração unindo técnicas de processamento de imagens e inteligência artificial. O processo de operação do software é dividido em algumas etapas distintas, as quais englobam algumas técnicas de processamento de imagem e inteligência artificial.

AQUISIÇÃO E CONSULTA

As etapas de aquisição da imagem e a consulta à base de dados não são abordadas neste trabalho. A aquisição depende de equipamentos físicos e a consulta depende de acesso aos órgãos relacionados.

O resultado final do reconhecimento está ligado diretamente à qualidade da imagem captada pela câmera de monitoramento e problemas como posicionamento, iluminação, reflexo, resolução, contraste, foco e outros relacionados à fotografia, podem dificultar ou, até mesmo, impedir o reconhecimento correto dos caracteres da placa.

PROCESSAMENTO DE IMAGEM

“Uma imagem digital pode ser considerada como uma representação discreta de dados que processam informação espacial (*layout*) e de intensidade (cor)” (SOLOMON *et al.*, 2013, p. 1). Esta representação pode tanto real como abstrata e permite o processamento, transmissão e armazenamento em dispositivos eletrônicos. O modelo de imagem digital utiliza neste trabalho, conhecido como *bitmap* ou mapa de bits, é o mesmo modelo utilizado em fotografias de câmeras digitais. “Os *bitmaps* nada mais são do que o conjunto de pixels da imagem digital, onde o valor de cada pixel representa as suas características de luminosidade e cor. Desse modo, os dados mapeiam a imagem, dando origem ao nome *bitmap*” (MARQUES *et al.*, 1999, p. 243). Cada um destes pixels utiliza uma quantidade específica de bits para representá-lo de acordo com o padrão adotado. Em imagens no padrão RGB, do inglês *Red*, *Green* e *Blue*, são utilizados 8 bits (1 byte) para representar cada uma das três camadas de cores. Então, para cada pixel deste padrão de imagem, são necessários 24 bits para representá-lo (SOLOMON *et al.*, 2013, p. 1). Uma imagem não compactada de tamanho VGA, do inglês *Video Graphics Array*, de dimensões de 640 pixels de largura por 480 pixels de altura, necessita de um espaço 921.600 bytes para armazenamento, sem considerar o tamanho do cabeçalho utilizado.

Segundo Marques *et al.* (1990, p. 1) o processamento de imagens pode ser aplicado na “análise automática por computador de informações extraídas de uma cena” e no “aprimoramento de informações pictóricas para interpretação humana”. Neste trabalho, o processamento consiste em manipular cada um dos pixels da imagem original, criar uma nova imagem e retirar somente as partes de interesse que, neste caso, é a placa do veículo e os seus caracteres correspondentes.

Em termos computacionais, este é o processo mais difícil e mais caro. Todas as demais etapas dependem diretamente da qualidade final desta e, como já dito, esta etapa depende diretamente da qualidade da imagem adquirida pelo equipamento de captura (câmera).

PRÉ-PROCESSAMENTO

A etapa de pré-processamento se divide em dois processos: escala de cinza e limiarização.

O primeiro processo consiste em transformar as três camadas de cores da imagem original colorida (RGB) em uma única camada, denominada escala ou níveis de cinza. De acordo com Solomon *et al.* (2013, p. 12) “a imagem em escala de cinza é, basicamente, uma soma ponderada dos canais de cor vermelho, verde e azul”. Na escala de cinza, cada pixel da imagem recebe um valor que varia em uma escala de 256 níveis (8 bits), de 0 à 255. O valor do nível 0 identifica a cor preta e o valor do nível 255 identifica a cor branca, qualquer outro valor entre 0 e 255 receberá uma tonalidade cinza variando de preto para branco. Para determinar a tonalidade de cinza de cada pixel calcula-se a sua luminosidade através de uma média ponderada das três cores que representam o pixel original: 29,89% do vermelho (*red*), 58,7% do verde (*green*) e 11,4% do azul (*blue*). Após o processo, é gerada uma nova imagem acinzentada, sem qualquer distinção de cor.

O segundo processo é a limiarização da imagem, também conhecido como binarização, que consiste em dividir a imagem em apenas dois tons: preto e branco (MARQUES *et al.*, 1999, p. 243). O processo é realizado utilizando um valor de limiar (*threshold*), o qual é um valor ideal para a divisão da imagem. Existem diversas técnicas para o cálculo do limiar, cada uma para uma aplicação específica. Para minimizar o consumo de processamento, neste trabalho a limiarização ocorre de duas formas diferentes com diferentes resoluções.

Para encontrar o posicionamento da placa do veículo utiliza-se a imagem original redimensionada proporcionalmente para largura ou altura máxima de 1024 pixels. A limiarização é executada com um valor de limiar fixo, inicialmente 127, qualquer valor de pixel acima deste limiar será considerado branco e valor igual ou abaixo será considerado preto. Caso o processo de extração de características, explicado adiante, falhe, outras tentativas de limiarização são executadas utilizando limiares diferentes variando em 32 tons para cima ou para baixo. Se todas as tentativas falharem, o processo de reconhecimento é interrompido, pois a imagem pode estar fora dos padrões mínimos exigidos para o reconhecimento.

Para a extração dos caracteres, recorta-se parte da imagem original utilizando as coordenadas previamente encontradas no processo anterior, redimensionando proporcionalmente para 512 pixels (1024 dividido 2) e executando novamente os processos de escala de cinza e limiarização. Nesta parte, para esta limiarização utiliza-se um limiar calculado pela média simples, somando todos os tons de cinza da imagem e dividindo o resultado pela quantidade total de pixels (largura x altura).

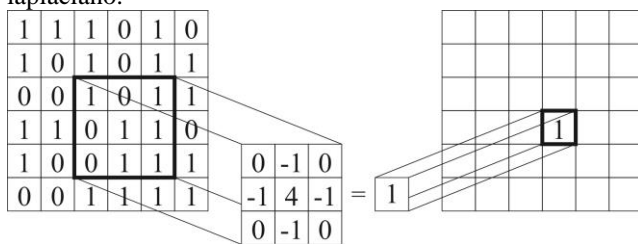
SEGMENTAÇÃO

“Segmentação é o nome dado ao processo genérico de subdividir uma imagem nas regiões ou objetos que a compõem” (SOLOMON *et al.*, 2013, p. 236). Na etapa de segmentação é realizada a detecção das bordas contidas na imagem, que são as variações brutas de intensidade luminosa e descontinuidade ou gradiente e que, geralmente, são as fronteiras ou contornos dos objetos que compõe a imagem. A detecção das bordas é realizada através de um operador diferencial de segunda ordem conhecido como laplaciano (Eq. 1).

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (1)$$

Para facilitar a implementação, o processo é realizado aplicando uma máscara de convolução (Fig. 1) que gera uma nova imagem com somente as bordas das áreas de interesse ou objetos da imagem original com fundo preto.

Figura 1 - Aplicação da máscara de convolução do filtro laplaciano.



No processo de convolução do filtro, os valores de todos os pixels da imagem original e seus vizinhos (3 x 3) são multiplicados com os valores correspondentes na máscara, a soma dos valores cada pixel e sua vizinhança geram um novo pixel que é transportado para uma nova imagem de tamanho igual à imagem original.

EXTRAÇÃO DE CARACTERÍSTICAS

A última parte do processamento de imagem, a extração de características, ou melhor, a extração da placa do veículo e dos caracteres, é realizada através de uma técnica similar ao crescimento de regiões. Solomon *et al.* (2013, p. 239) diz que “o crescimento de regiões é uma abordagem à segmentação em que os pixels são agrupados em regiões cada vez maiores, seguindo algum critério de similaridade pré-definido”. Neste processo, o primeiro pixel preto, ou zero, da imagem de bordas é selecionado e adicionado em um vetor com as suas respectivas coordenadas (x e y). Os quatro vizinhos adjacentes (norte, leste, oeste e sul) do pixel são verificados, se os valores dos mesmos forem iguais a zero eles são, também, adicionados ao vetor e os seus vizinhos correspondentes são verificados em um processo iterativo até que todos os pixels da região sejam encontrados. Para cada vizinhança

de pixels é obtido um vetor com todos os pixels vizinhos iguais a zero. O processo é repetido até que não seja encontrado nenhum outro pixel na imagem com valor igual a zero.

Figura 2 - Etapas da segmentação: (a) imagem original, (b) limiarizada e (c) segmentada.

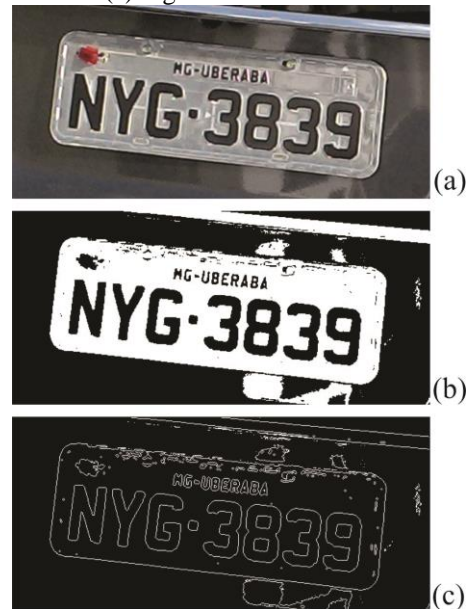
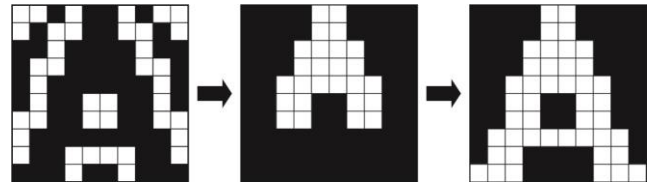


Figura 3 - Exemplificação do processo de extração de características.



No fim do processo obtém-se uma matriz contendo todos os objetos separados pelas bordas encontradas no processo de segmentação.

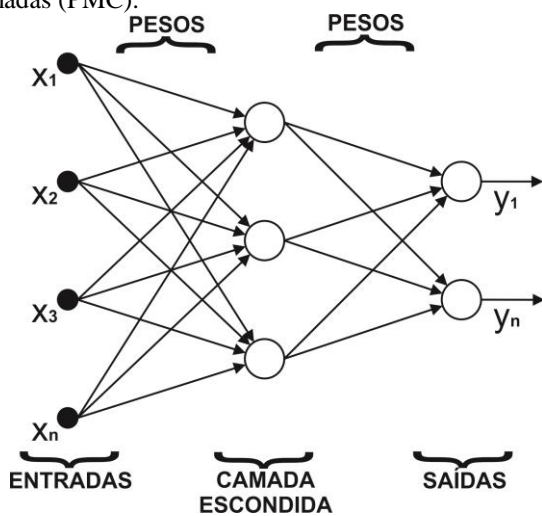
Para determinar qual destes objetos são os possíveis caracteres da placa do veículo, é aplicado um filtro comparando o tamanho e posicionamento dos mesmos, os que tiverem tamanhos fora do padrão, desproporcionais ou distantes são descartados até que restem somente sete objetos.

Utilizando as coordenadas dos objetos restante da matriz, é recordada uma nova imagem a partir da imagem original (colorida) e, nesta imagem, é realizado todo o processo novamente, da escala de cinza até a extração de características. Aplica-se novamente o filtro nos objetos encontrados, se forem encontrados sete objetos de altura e distância similar, o processo de segmentação é finalizado com sucesso.

RECONHECIMENTO DOS CARACTERES

Para o reconhecimento dos caracteres da placa são utilizadas redes neurais artificiais com a arquitetura conhecida como *perceptron* múltiplas camadas (PMC). As redes PMC possuem pelo menos uma camada de neurônios escondida, situada entre a camada de entrada e a camada de saída. Uma rede PMC é capaz realizar aprendizado e reconhecer padrões entre outras aplicações. Segundo Silva *et al.* (2010, p. 91) “as redes PMC são ainda caracterizadas pelas elevadas possibilidades de aplicações em diversos tipos de problemas relacionados com as mais diferentes áreas do conhecimento”.

Figura 4 - Ilustração da rede *perceptron* múltiplas camadas (PMC).



As redes neurais artificiais (RNAs) são modelos computacionais inspirados no funcionamento básico do cérebro biológico. Uma rede é composta por vários “neurônios” interconectados entre si. Cada neurônio recebe diferentes valores em suas camadas de entrada e cada valor é ponderado utilizando pesos previamente ajustados, todos os resultados são acumulados e transferidos para a camada posterior através de uma função de ativação. No PMC, as funções de ativação mais comumente usadas são: a tangente hiperbólica e a logística sigmoide. Neste trabalho, todos os valores, tanto de entrada como de saída, são do tipo binário (zero e um), portanto a função logística foi escolhida porque ela sempre resulta em valores reais entre zero e um. A expressão matemática da função logística é dada pela equação 2.

$$g(u) = \frac{1}{1 + e^{-u}} \quad (2)$$

Este trabalho utiliza duas abordagens diferentes de topologia das redes PMC, a primeira utiliza uma única rede capaz de reconhecer os caracteres numéricos e alfabéticos, a segunda utiliza duas redes distintas: uma para reconhecer exclusivamente os caracteres numéricos e outra para os caracteres alfabéticos. Na segunda abordagem além de

melhorar de maneira significativa a precisão no reconhecimento dos números, há também uma diminuição no processamento durante a operação das redes, visto que 4 dos 7 dígitos verificados são sempre numéricos e esta rede possui menos neurônios que as outras redes.

TREINAMENTO DA REDE NEURAL

O processo de treinamento da rede PMC é do tipo supervisionado. “A estratégia de treinamento supervisionado consiste em se ter disponível, considerando cada amostra dos sinais de entrada, as respectivas saídas desejadas [...]” (SILVA *et al.*, 2010, p. 51). Quanto maior o número de amostras apresentadas durante o processo de treinamento, melhores serão os resultados na execução posterior da rede.

Neste trabalho, devido à dificuldade em adquirir imagens de placas reais suficientes para o treinamento, as amostras dos caracteres foram criadas automaticamente através do software com diferentes ângulos de inclinação variando de -15° à $+15^\circ$. Esta variação angular foi necessária devido aos diferentes padrões de posicionamento das placas nas imagens utilizadas para os testes. A resolução N° 241 de 22 de julho de 2007 do CONTRAN (Conselho Nacional de Trânsito) determina as especificações técnicas para as placas de identificação e uma destas especificações é que a tipologia das placas e tarjetas deve seguir o modelo da fonte Mandatory (Fig. 5). Portanto, todas as amostras foram geradas utilizando esta fonte e todas as amostras utilizadas nos testes foram de placas utilizando esta tipologia.

Figura 5 - Caracteres da fonte Mandatory.



Foram geradas 31 imagens diferentes para cada caractere perfazendo um total de 1116 amostras, sendo 310 números e 806 letras. Cada caractere (número e letra) é representado pelo seu valor binário da tabela ASCII (American Standard Code for Information Interchange) utilizando um conjunto de sete bits de saída. As imagens geradas possuem tamanho fixo de 10 pixels de largura por 15 pixels de altura (Fig. 6) e, através destas imagens, é gerado um vetor de 150 valores binários. Deste modo, cada uma das amostras possuem 150 valores de entrada e 7 valores de saída, todos conjuntos de zeros e uns.

Figura 6 - Exemplos de amostras da letra “A”.



O treinamento da rede PMC é realizado através de uma variação do algoritmo back-propagation, onde cada amostra é apresentada a rede e o valor de calculado e comparado com o valor de saída desejado. O erro encontrado é propagado de modo reverso para os neurônios das camadas anteriores até a camada de entrada.

O processo é repetido sucessivamente até que o erro médio global da rede seja menor do que a precisão estabelecida para o treinamento. Cada repetição deste processo, onde todas as amostras são apresentadas à rede uma vez, é chamado de época e a quantidade de épocas varia de acordo com diversos fatores como precisão, taxa de aprendizagem, número de camadas e neurônios, complexidade das amostras apresentadas e valores iniciais dos pesos. Não existem regras específicas de como treinar a rede, Silva *et al.* (2010, p. 52) diz que “o treinamento supervisionado é um caso típico de inferência indutiva pura”.

RESULTADOS

Para a realização dos testes foram utilizadas 83 imagens diferentes de veículos com diversas posições, resoluções e cores.

Como observado na figura 5, os caracteres numéricos “1” e “0” são idênticos as letras “I” e “O”, por isso na primeira abordagem, onde foi utilizado uma única rede PMC, foi necessário acrescentar um novo valor de entrada na rede, valor zero (0) quando se tratar de números e um (1) quando se tratar de letras. Na abordagem de duas redes isto não foi necessário já que as redes separam totalmente os números das letras.

Para padronizar os testes e os resultados, os treinamentos de todas as redes foram executados utilizando os mesmos pesos iniciais criados aleatoriamente no primeiro treinamento, caso contrário os mesmos parâmetros iniciais produziram diferentes resultados para os mesmos valores de testes.

Durante a execução dos testes notou-se a necessidade de implementar uma variação no algoritmo de treinamento a fim de diminuir o tempo gasto na convergência das redes. Foi utilizada uma taxa de aprendizagem dinâmica capaz de variar de acordo com o histórico do próprio treinamento ajustando-se automaticamente durante o processo. Inicialmente a taxa é ajustada para 0,1 e se após 10 épocas consecutivas o erro global da rede diminuiu conforme o esperado então a taxa de aprendizagem é ajustada com o acréscimo de 10%, mas se em qualquer época o erro global da rede aumentou, a taxa é ajustada com o decréscimo de 10%. Com esta taxa

variável foi possível atingir precisão de até 1×10^{-16} com facilidade.

No treinamento da rede única foram necessárias duas camadas de neurônios escondidas, a primeira com 100 e a segunda com 40 neurônios. Em um treinamento com precisão de 1×10^{-9} a rede foi capaz de reconhecer 100% dos 581 caracteres (letras e números) testados.

Tabela 1 - Treinamento da rede única.

| Camadas Escondidas | Épocas | Tempo | Erro Máximo | Nível de Acerto |
|--------------------|--------|----------|---------------------|-----------------|
| 100 e 20 | 516 | 00:00:50 | 1×10^{-3} | 97,93% |
| 100 e 20 | 1812 | 00:02:57 | 1×10^{-6} | 99,83% |
| 100 e 20 | 3121 | 00:04:55 | 1×10^{-9} | 99,48% |
| 100 e 20 | 4428 | 00:06:50 | 1×10^{-12} | 99,14% |
| 100 e 30 | 463 | 00:00:53 | 1×10^{-3} | 97,25% |
| 100 e 30 | 1763 | 00:03:15 | 1×10^{-6} | 98,62% |
| 100 e 30 | 3068 | 00:05:44 | 1×10^{-9} | 99,14% |
| 100 e 30 | 4375 | 00:08:04 | 1×10^{-12} | 99,31% |
| 100 e 40 | 196 | 00:00:54 | 1×10^{-3} | 98,11% |
| 100 e 40 | 1485 | 00:03:25 | 1×10^{-6} | 99,48% |
| 100 e 40 | 2785 | 00:05:44 | 1×10^{-9} | 100,00% |
| 100 e 40 | 4097 | 00:08:11 | 1×10^{-12} | 100,00% |

Fonte - Dados obtidos pelo autor.

A rede alfabética ficou com a mesma topologia da rede única, duas camadas de neurônios escondidas, 100 e 40 neurônios respectivamente. E também, de modo semelhante a outra rede, com precisão de 1×10^{-9} a rede foi capaz de reconhecer 100% dos 249 caracteres alfabéticos testados.

Tabela 2 - Treinamento da rede alfabética.

| Camadas Escondidas | Épocas | Tempo | Erro Máximo | Nível de Acerto |
|--------------------|--------|----------|---------------------|-----------------|
| 100 e 20 | 299 | 00:00:20 | 1×10^{-3} | 99,60% |
| 100 e 20 | 1582 | 00:02:00 | 1×10^{-6} | 99,60% |
| 100 e 20 | 2883 | 00:03:36 | 1×10^{-9} | 99,60% |
| 100 e 20 | 4183 | 00:05:02 | 1×10^{-12} | 98,80% |
| 100 e 30 | 263 | 00:00:20 | 1×10^{-3} | 97,99% |
| 100 e 30 | 1563 | 00:02:11 | 1×10^{-6} | 99,20% |
| 100 e 30 | 2867 | 00:03:56 | 1×10^{-9} | 99,20% |
| 100 e 30 | 4172 | 00:05:25 | 1×10^{-12} | 99,60% |
| 100 e 40 | 196 | 00:00:17 | 1×10^{-3} | 98,39% |
| 100 e 40 | 1485 | 00:02:14 | 1×10^{-6} | 99,20% |
| 100 e 40 | 2785 | 00:04:09 | 1×10^{-9} | 100,00% |
| 100 e 40 | 4097 | 00:05:40 | 1×10^{-12} | 100,00% |

Fonte - Dados obtidos pelo autor.

No treinamento da rede numérica, com uma única camada escondida de 20 neurônios e precisão de 0,00001 a rede foi capaz de reconhecer 100% dos 332 caracteres numéricos testados.

Tabela 3 - Treinamento da rede numérica.

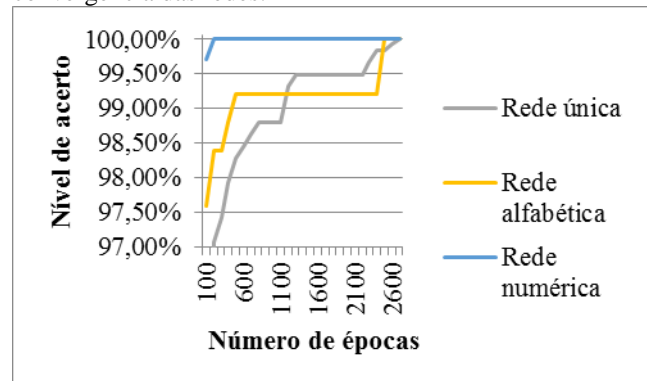
| Camada Escondida | Épocas | Tempo | Erro Máximo | Nível de Acerto |
|------------------|--------|----------|---------------------|-----------------|
| 10 | 363 | 00:00:01 | 1×10^{-3} | 99,10% |
| 10 | 1659 | 00:00:04 | 1×10^{-6} | 98,80% |
| 10 | 2963 | 00:00:09 | 1×10^{-9} | 98,80% |
| 10 | 4268 | 00:00:14 | 1×10^{-12} | 98,80% |
| 15 | 297 | 00:00:01 | 1×10^{-3} | 99,70% |
| 15 | 1596 | 00:00:00 | 1×10^{-6} | 99,40% |

| Camadas Escondidas | Épocas | Tempo | Erro Máximo | Nível de Acerto |
|--------------------|--------|----------|---------------------|-----------------|
| 15 | 2903 | 00:00:14 | 4×10^{-9} | 98,80% |
| 15 | 4203 | 00:00:19 | 9×10^{-12} | 99,10% |
| 20 | 290 | 00:00:01 | 1×10^{-3} | 100,00% |
| 20 | 1583 | 00:00:11 | 1×10^{-6} | 100,00% |
| 20 | 2888 | 00:00:20 | 1×10^{-9} | 100,00% |
| 20 | 4198 | 00:00:27 | 7×10^{-12} | 100,00% |

Fonte - Dados obtidos pelo autor.

De acordo com os dados apresentados, indiferentemente da abordagem escolhida, todos os caracteres das imagens utilizadas nos testes foram reconhecidos com precisão. A diferença principal encontrada foi a quantidade necessária de épocas para treinamento (Graf. 1), porém em uma aplicação real isto pode ser irrelevante já que uma rede neural é treinada somente uma vez ou em mudança proposital do conjunto de amostras de treinamento.

Gráfico 1 - Comparativo de quantidade de épocas para convergência das redes.



Fonte - Dados obtidos pelo autor.

CONCLUSÃO

Conforme é possível observar com os resultados obtidos, o aplicativo apresentou excelente desempenho no reconhecimento dos caracteres, independente da abordagem utilizada. Em ambos os tipos, numéricos e alfabéticos, o treinamento atingiu 100% de eficiência nas 83 placas de amostras utilizadas para testes. Todos os ensaios apresentados foram executados em ambiente de testes, porém as imagens utilizadas não sofreram nenhum tipo de manipulação. Em uma aplicação real, desde que as imagens possuam qualidade similar das utilizadas nos testes, os mesmos níveis de acertos podem ser alcançados.

Conforme o propósito principal deste trabalho, devido a simplicidade do desenvolvimento, operação e manutenção, este sistema pode facilmente ser instalado em sistemas de vigilância, sistemas de fiscalização, controle de entradas e saídas de estacionamentos públicos ou privativos, entre outras aplicações.

REFERÊNCIAS

SOLOMON, C.; BRECKON, T. **Fundamentos de Processamento Digital de Imagens: Uma Abordagem Prática com Exemplos em Matlab.** Rio de Janeiro: LTC, [2013]. 289 p.

MARQUES FILHO, O.; VIEIRA NET, H. **Processamento Digital de Imagens.** Rio de Janeiro: Brasport. [1999].

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais: para Engenharia e Ciências Aplicadas - Curso Prático.** São Paulo: Artliber, [2010]. 399 p.

BRASIL. Federal. Decreto-Lei nº 237, de 28 de Fevereiro de 1967. Modifica o Código Nacional de Trânsito. Brasília : 1967.

BRASIL. Conselho Nacional de Trânsito. **Resolução nº 241, de 22 de Junho de 2007. Especificações Técnicas para as Placas de Identificação de Veículos.** Brasília : 2007.