



Classificação de Vagas de Estacionamento com Utilização de Rede Perceptron Multicamadas

Heráclito Santos Falcão, Agnaldo Volpe Lovato, Alex Ferreira dos Santos, Lucas Santos de Oliveira, Robson Hebraico Cipriano Maniçoba, Marcelo Alves Guimarães e Murilo Silva Santana

Departamento de Química e Exata (DQE). Universidade Estadual do Sudoeste da Bahia (UESB), Jequié, Bahia, Brasil.

Resumo—Este projeto tem como objetivo o desenvolvimento de um protótipo para identificação de vagas em estacionamentos abertos onde se tenha boa visão aérea, sem obstáculos, possibilitando que sejam identificadas vagas ocupadas e livres. Para isso, foram utilizadas técnicas de processamento de imagens e reconhecimento de padrões a partir de Redes Neurais Artificiais (RNAs). Para auxiliar na implementação do protótipo dos algoritmos, foi utilizada a ferramenta computacional Matlab (do inglês, “MatrixLaboratory”). Foi necessária a confecção de uma maquete para simular o ambiente de um estacionamento, com o objetivo de colocar em prática a aquisição de imagens realizada pela webcam, o processamento destas, o treinamento da rede neural, a classificação das vagas e por fim, a exibição dos resultados. Os resultados apresentam a possibilidade de utilizar o reconhecimento de padrões por meio da captura de imagens na classificação de vagas em estacionamentos.

Palavras chave—Inteligência Artificial, Redes Neurais, Reconhecimento de Padrões.

I. INTRODUÇÃO

Com o aumento da renda das classes sociais brasileiras e até mesmo com a necessidade de locomoção mais rápida das pessoas, é notável um crescimento do número de carros nas cidades. O aumento do número de automóveis está levando médias e grandes cidades a uma rápida asfixia [1]. Segundo o Denatran (Departamento Nacional de Trânsito), a frota de veículos cresceu 119% em dez anos no Brasil, o país fechou o ano de 2010 com 64.817 milhões de veículos registrados (Figura 1).

Diante das diversas mudanças sociais e pelo significativo aumento do número de carros nas cidades, uma característica chama bastante atenção: a necessidade de encontrar vagas para estacionamento.

Atualmente não foi encontrada uma solução tão eficaz para resolver esse tipo de demanda, principalmente nos centros urbanos, onde se concentram os maiores números de automóveis. Para os condutores de veículos que cotidianamente precisam, por alguma razão, deslocar-se ao tumultuado centro da cidade, encontrar uma vaga para

estacionamento é em geral um motivo causador de estresse [2]. Isso não ocorre apenas nos grandes centros urbanos. Até em momentos de lazer, achar uma vaga se torna uma missão torturante e desanimadora. Ao longo da orla é difícil encontrar vagas e muitos motoristas recorrem a estacionamentos irregulares ou a flanelinhas [3]. Algumas pessoas já viabilizaram alguns tipos de soluções, como por exemplo, criar estacionamentos subterrâneos. Algo muito caro para a realidade de muitos lugares. No plano para acabar com a falta de vagas na área central de Brasília e na Esplanada dos Ministérios, pensam-se na construção de estacionamentos subterrâneos [4].

Evolução do total da frota na década

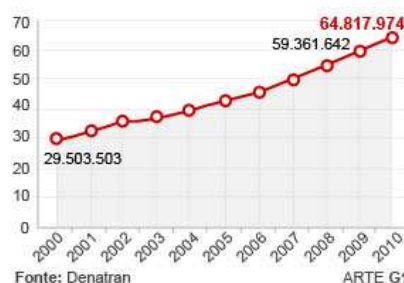


Figura 1: Evolução do total da frota de veículos na década. Fonte: Denatran

Atualmente, existem softwares que utilizam sensores com ultrassom para detecção de vagas. Esta técnica consiste na instalação de sensores de presença em cada uma das vagas em um estacionamento, na qual fornece o número de vagas disponíveis em cada ala ou andar do estacionamento. No entanto, seu alto custo inviabiliza a instalação nos centros urbanos uma vez que o Brasil ainda é um país em desenvolvimento.

Pesquisas realizadas indicam que as maiores causas de congestionamentos nas cidades ocorrem por conta de condutores em busca de vagas para estacionar. A procura de estacionamento provoca congestionamentos e o cometimento de infrações, como filas duplas e estacionamento sobre os passeios [5].

Atualmente, com a popularização de equipamentos capazes de gerar, transmitir e armazenar vídeo com custos bastante acessíveis é possível obter informações dinâmicas ou estáticas (vídeo ou foto) de forma eficiente. Logo a possibilidade de processamento dessas imagens e a classificação de uma vaga de estacionamento se torna uma realidade.

A Inteligência Artificial (IA) ocupa a área de sistematização e automação de tarefas, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana [6]. Dentro dessa área existe as Redes Neurais Artificiais (RNA) que segundo Braga [7], esta forma de computação não-algorítmica é caracterizada por sistemas que, em algum nível, relembra a estrutura do cérebro humano.

As RNAs são constituídas por neurônios (unidade de processamento simples), que têm a função de calcular determinadas funções matemáticas (normalmente não-lineares). Estas unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos, estas conexões estão associadas a pesos, que armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede [14].

As redes neurais constituem um dos ramos da inteligência artificial mais bem sucedidos, na qual o campo de aplicação é bastante amplo. De acordo com Braga [7], a solução através de RNAs é bastante atrativa.

Este projeto apresenta uma forma de utilizar imagens capturadas por meio de uma câmera para auxiliar na identificação de vagas livres e ocupadas de um estacionamento, para tal será desenvolvida uma rede neural capaz de auxiliar na classificação do estado das vagas, visto sua capacidade de resolução e generalização de problemas.

I. FUNDAMENTOS TEÓRICOS

A. Processamento de imagem

Segundo Marques Filho [8], o processamento de imagens digitais envolve procedimentos normalmente expressos sob forma algorítmica. Neste sentido a maioria dessas funções de processamento de imagens são manipuladas por computadores.

Em [9] a autora entende o Processamento Digital de Imagens como a manipulação de uma imagem por computador de modo que a entrada e a saída do processo sejam imagens. A finalidade em utilizar essa técnica é melhorar certas imperfeições (tais como ruídos, contrastes e/ou brilho inadequados) a fim de aprimorar a aquisição da imagem para etapas subsequentes.

Em [8] ressalta que o processamento de imagem viabiliza grande número de aplicações em duas categorias bem distintas: (1) o aprimoramento de informações pictóricas para interpretação humana; e (2) a análise automática por computador de informações extraídas de uma cena.

A imagem digital nada mais é que uma matriz na qual as linhas e colunas identificam um ponto na imagem. Uma imagem digital é uma função $f(x,y)$ discretizada em amplitude e no espaço [8].

As palavras *picture element* (pixel) se referem a menor unidade que constitui uma imagem digital. Para formar uma imagem completa é necessário um conjunto de vários (ou milhares) de pixels. Na Figura 2, é possível notar a representação de uma imagem com 256 tons de cinza.



Figura 2: Representação de uma imagem ampliada de 10x10 com 256 tons de cinza [10].

Posterior à aquisição da imagem, é necessária a utilização de técnicas de processamento de imagem para assim melhorá-la, o que aumenta a chance de sucesso dos processos seguintes [11]. O processamento de imagem consiste em aplicar sobre a imagem transformações necessárias para facilitar a extração das informações contidas nela para uma aplicação específica. As técnicas utilizadas são classificadas em duas categorias principais. Uma é referente ao domínio espacial e a outra referente à manipulação direta dos pixels de uma imagem.

Nesta última técnica, as propriedades da imagem seguem três fatores principais; primeiro a localização do ponto a ser analisado, definido por linha e coluna; segundo a resolução radiométrica que o sistema sensor possui; e por fim a resolução espacial, definida pelo tamanho do pixel.

B. Redes Neurais Artificiais

A Rede Neural Artificial (RNA) é um dos ramos da Inteligência Artificial que Braga [12] afirma ser uma forma não-algorítmica, é caracterizada por sistemas que, em algum nível, relembra a estrutura do cérebro humano. Além disso, como a computação neural não é baseada em regras, então ela se constitui em uma alternativa à computação algorítmica convencional.

As RNAs têm a capacidade de calcular, armazenar e distribuir as informações na rede interligada. São sistemas distribuídos compostos por unidades de processamento simples (neurônios artificiais) que calculam determinadas funções matemática (normalmente não-lineares). Tais unidades são dispostas em uma ou mais camadas e interligadas por grande número de conexões, geralmente unidirecional [7]. Braga [12] aponta que na maioria dos modelos, essas conexões estão associadas a pesos, que têm a função de armazenar o conhecimento adquirido pelo modelo e servem para ponderar a entrada recebida por cada neurônio da rede.

Em [13] afirma que uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso.

As etapas de uma RNAs para solucionar um conjunto de problemas têm sua fase inicial no aprendizado, onde um

conjunto de exemplos é apresentado para a rede, com isso, a rede extrai as características do problema. Com base nessas características é possível nas próximas etapas do processo reutilizá-las, a fim de encontrar possíveis respostas para determinados problemas.

A capacidade de aprender por meio de exemplos e de generalizar a informação aprendida é, sem dúvida, o atrativo principal da solução de problemas por meio das RNAs. Complementando, as RNAs têm a capacidade de extrair informações apresentadas de forma não explícita através de exemplos [13].

Neurônios individuais possuem capacidade computacional limitada, porém quando existe um conjunto de neurônios artificiais conectados formando uma rede, então essa capacidade se eleva consideravelmente, sendo capaz de resolver problemas de alta complexidade.

Santiago Ramon y Cajal (1911) instituiu o neurônio como uma estrutura básica do cérebro, responsável por enviar e receber informações, atuando através da produção e transmissão de sinais elétricos, chamados impulsos nervosos. É composto basicamente pelos dendritos e axônios, que recebem e transmitem respectivamente sinais elétricos a outros neurônios, e corpo celular (soma) responsável, pelo processamento destes sinais (Figura 3).

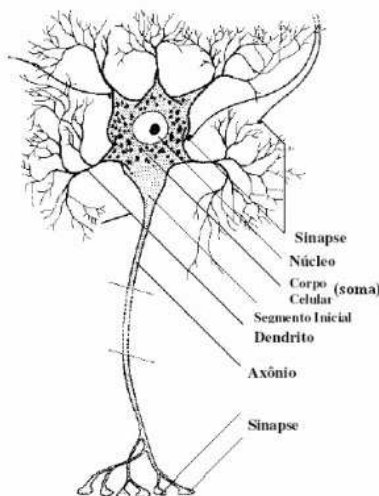


Figura 3: Esquema de um neurônio biológico [14].

Em 1943, McCulloch e Pitts propuseram matematicamente, de forma simplificada, o funcionamento de um neurônio biológico (Figura 4). Nesse modelo, o neurônio possui i entradas (equivalente aos dendritos) x_1, x_2, \dots, x_i e apenas uma saída (equivalente ao axônio) y . Para poder simular a sinapse, cada entrada do neurônio tem um peso w_1, w_2, \dots, w_i cujos valores podem ser positivos (excitatórios) ou negativos (inibitórios). Os pesos têm como finalidade armazenar o conhecimento e determinar a intensidade com que cada entrada contribuirá no resultado do neurônio. O corpo celular é emulado simplesmente somando os valores do produto de suas entradas com seus respectivos pesos ($x_i \cdot w_i$) e, se a soma for maior ou igual ao seu limiar (*threshold*), a sua saída y é

ativada com valor 1.

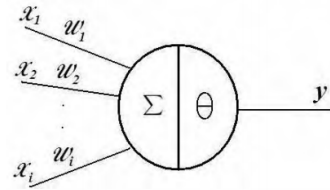


Figura 4: Modelo matemático de um neurônio biológico [7]

Nos anos seguintes, outros pesquisadores propuseram modificações ao modelo de McCulloch e Pitts, sendo que as principais mudanças foram realizadas na função que determina o estado de ativação da saída do neurônio (função ativação), onde teve o acréscimo de um *bias*.

Um neurônio artificial é composto por um conjunto de entradas $x_1, x_2, x_3, \dots, x_n$ que são os sinais vindos de outros neurônios, uma saída y_k , (Σ) somador que é a responsável pelo calculo efetivo da entrada, pesos que representam os pesos sinápticos das conexões entre os neurônios da camada anterior e da camada k , *bias* (B_k) que tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação (*bias* age sendo um peso extra, nas conexões das unidades cuja entrada é sempre um) e uma função de ativação $\phi(\cdot)$ (Figura 5).

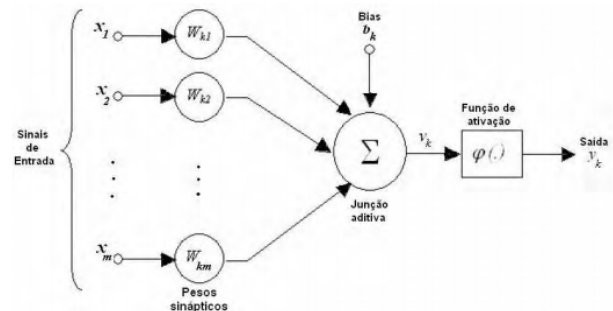


Figura 5: Modelo de neurônio base para projetos de RNA [13].

Em uma estrutura de uma RNA para determinado problema é necessário observar vários fatores [12]:

- Complexidade do problema;
- Dimensionalidade do espaço de entrada;
- Características dinâmicas ou estáticas;
- Conhecimento a priori sobre o problema;
- Representatividade dos dados;

Esses fatores vão influenciar diretamente a arquitetura ou topologia de um RNA.

Quanto ao número de camadas:

- Redes com camada única – forma mais simples de uma rede em camadas, surge quando se tem uma camada de entrada que se projeta para a camada de saída;
- Redes com múltiplas camadas – diferencia da rede de camada única, por causa da presença de uma ou mais camadas ocultas.

Tipos de conexões:

- *Feedforward* ou acíclica – a saída do neurônio na i -

ésima camada não pode ter entradas com neurônios em camadas de índice menor ou igual a i ;

- *Feedback* ou cíclica – a saída do neurônio na i -ésima camada tem entradas com neurônios em camadas de índice menor ou igual a i .

Quanto ao aprendizado das RNAs, [15] aponta que o conhecimento está armazenado nos pesos e na topologia. Os pesos simbolizam o conhecimento da rede e determinam a ponderação que determinada entrada possui.

No processo de aprendizado de uma rede neural artificial, existem basicamente duas maneiras: aprendizado supervisionado e aprendizado não-supervisionado. O tipo de aprendizagem é determinado pela maneira com a qual a modificação dos parâmetros ocorre [13].

O aprendizado supervisionado implica na existência de um supervisor ou professor externo, no qual fica responsável em fiscalizar e estimular as entradas da rede por meio de padrões e observar a saída, conferindo com a saída desejada.

Em [12] difunde que o aprendizado supervisionado se aplica a problemas em que se deseja obter um mapeamento entre padrões de entrada e saída. O exemplo mais conhecido de aprendizado supervisionado é a regra Delta e suas generalizações para rede de múltiplas camadas, como o algoritmo *back-propagation*.

No aprendizado não-supervisionado não existe professor ou supervisor externo para fiscalizar, somente os padrões de entradas estão disponíveis. Esse modelo é utilizado para problemas que visam à descoberta de características estatísticas relevantes nos dados de entrada, como a descoberta de agrupamentos, ou classes [14].

Seguindo a evolução das RNAs, na década de 1950 mais precisamente em 1958 com Frank Rosenblatt, o conceito de aprendizado foi introduzido nas RNAs. Surgiu assim o Perceptron, composto por uma estrutura de rede do modelo de McCulloch e Pitts, sendo a forma mais simples de uma RNA usada para classificação de padrões ditos linearmente separáveis, ou seja, padrões que se encontram em lados opostos de um hiperplano [13].

Em [12] explica que a topologia original era composta por unidade de entrada (retina), por um nível intermediário formado pelas unidades de associação e por um nível de saída formado pela unidade de resposta. No entanto, apesar dessa topologia possuir três níveis, ela é conhecida como *perceptron* de uma única camada, pelo motivo de que somente o nível de saída (unidade de resposta) apresenta propriedades adaptativas. A Retina são unidades sensoras, a unidade de associação, que embora sejam formadas por neurônios do modelo de McCulloch e Pitts (MCP), possuem pesos fixos, definidos no período de treinamentos.

Na Figura 6, existem cinco entradas representadas de x_1, \dots, x_5 , tem peso (sinapse) w_1, \dots, w_5 , w_0 (bias) na entrada do neurônio, d é a saída intermediária e O a saída ativada da rede.

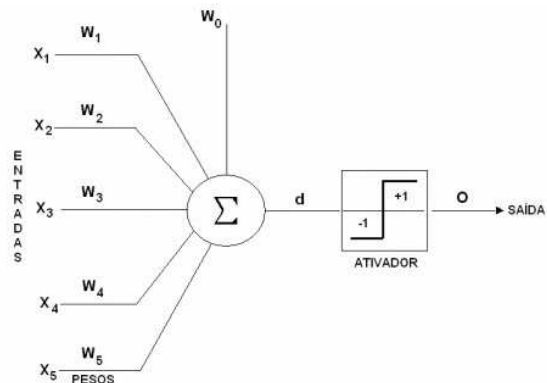


Figura 6: Modelo *Perceptron* [16]

Em [16] e [13] são discutidos as limitações do *Perceptron* para alguns problemas que não são linearmente separáveis. Uma possível solução seria aumentar o conjunto de treinamento para refinar mais os pesos obtidos, contudo, pode não haver uma divisão bem definida das classes, o que pode provocar erros de classificação, ou ainda, as características do problema considerado, não permite uma divisão clara entre duas ou mais classes, mesmo através de linhas bem definidas, ou seja, uma rede de única camada tem limitações de resolver apenas problemas com características lineares.

No entanto, o *Perceptron* de múltiplas camadas consegue resolver os problemas não-lineares, assim, como à maioria das situações e problemas reais são não-lineares, então ele se adequa melhor na resolução de problemas. O *Multilayer Perceptron* foi elaborado para resolver problemas complexos, no quais não seria possível resolver com modelo de neurônio básico.

À rede neural de múltiplas camadas composta por neurônios com funções sigmoidais nas camadas intermediárias dá-se o nome de *Perceptron* de Múltiplas Camadas (MLPs – *Multi-Layer Perceptron*) [12].

Basicamente uma rede MLP é constituída de uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Para treinar esse tipo de rede utiliza-se o algoritmo *backpropagation* [15].

Na Figura 7 se encontra a arquitetura de uma rede MLP com uma camada de entrada composta por três neurônios, duas camadas ocultas, na qual cada uma possui quatro neurônios, e uma saída, com dois neurônios. Observa-se que no modelo, qualquer neurônio em qualquer camada da rede se encontra conectado a todos os neurônios das camadas anteriores. É empírica definição do número de camadas escondidas e quantidade de neurônios em cada camada.

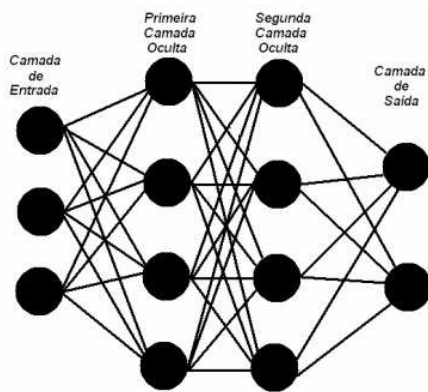


Figura 7: Arquitetura de RNA MLP com duas camadas ocultas

Os *perceptrons* de múltiplas camadas têm sido aplicados com sucesso para resolver diversos problemas difíceis, através do seu treinamento de forma supervisionada com retropropagação de erro [13]. Exemplos de aplicabilidade: reconhecimento de caracteres, previsão do comportamento de ações na bolsa, verificação de assinaturas, segurança em transações com cartões de crédito, diagnóstico médico entre outros [7].

Para treinar as redes de *Perceptron* de Múltiplas Camadas é utilizado o algoritmo de *backpropagation* que consiste de duas fases: a propagação (*forward*) e a retropropagação (*backward*) de um conjunto de sinais através da rede. Na etapa de propagação existe a aquisição dos dados através da camada de entrada e sua propagação através de toda a rede, gerando uma saída. Então a saída é comparada com a saída desejada e um valor de erro é calculado. Depois disso, inicia o processo de retropropagação, esse erro é propagado de volta à rede neural e utilizado para ajustar os pesos, buscando assim reduzir o máximo possível os erros a cada iteração para que o resultado se aproxime da saída desejada (Figura 8).

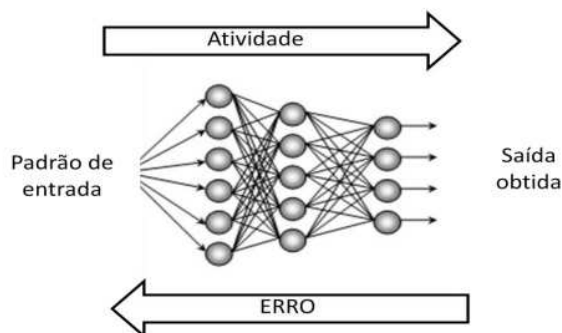


Figura 8: Esquema do funcionamento do algoritmo *Backpropagation*

A maneira como os neurônios estão organizados em uma rede neural está estritamente ligado em qual tipo de problema será solucionado, com isso a arquitetura de redes neurais em geral se dividem em três grupos. Redes *feedforward* de uma única camada, redes *feedforward* de múltiplas camadas e redes recorrentes.

II. EXECUÇÃO DO PROJETO

Para o desenvolvimento deste projeto foram utilizados: webcam, computador, maquete de madeira, carros em miniatura e uma ferramenta de simulação. A Figura 9 apresenta a maquete e seus componentes.



Figura 9 - Maquete do estacionamento

Inicialmente foram pesquisadas funções existentes no Matlab para auxiliar na captura e processamento das imagens, em seguida foi analisado o melhor modelo de rede neural, optando-se pela rede MLP (*Multi-Layer Perceptron*) devido ao seu poder de generalização e tratamento de problemas cujo *Perceptron* simples não trata. O próximo passo foi definir a arquitetura da rede, modelando a quantidade de neurônios de entrada, saída e número de camadas escondidas. Posteriormente, foi realizada a aquisição de dados, treinamento da rede e teste de treinamento, sendo este processo repetido toda vez que a rede não apresentava os resultados esperados (Figura 10).

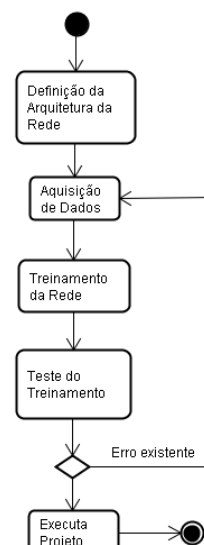


Figura 10 - Processo de Treinamento

Todo o desenvolvimento foi realizado utilizando a ferramenta Matlab e sua biblioteca disponível para modelagem e execução de redes neurais, as seções que seguem apresentam em detalhes estas etapas.

A. Definição da Arquitetura da Rede

Para definir a arquitetura da rede, é importante inicialmente analisar quais serão os dados de entrada e saída. Na entrada, as informações apresentadas representarão parte da imagem capturada pela webcam, as quais representam os dados correspondentes a uma única vaga de estacionamento. Na Figura 11, pode ser observado o retângulo verde, que representa a vaga capturada para o treinamento. Cada vaga possui um tamanho de 39 x 24 pixels, que será convertido em uma matriz unidimensional com 936 linhas, que serão utilizadas para a entrada da rede. A saída será representada por apenas 1 neurônio que informará 0 (zero) caso a vaga esteja livre e 1(um) quando estiver ocupada.

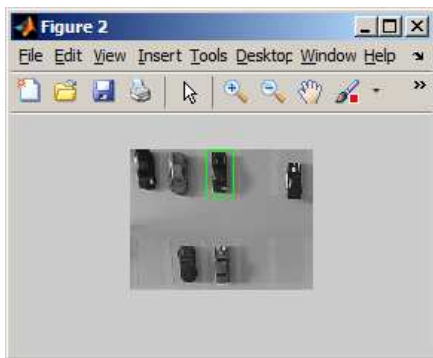


Figura 11 - Imagem do Estacionamento com o corte de uma vaga

A definição do número de camadas escondidas e quantidade de neurônios em cada camada realiza-se de forma empírica. Estudos realizados comprovam que uma única camada escondida na rede MLP consegue solucionar problemas de classificação. Como o problema em questão trata-se de uma classificação de padrões, optou-se pelo uso de uma única camada, atribuindo a mesma 5 neurônios, que foram suficientes para o treinamento (Figura 12).

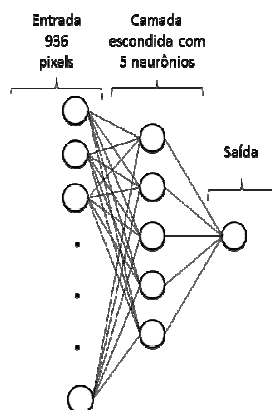


Figura 12 - Arquitetura da Rede

B. Aquisição de Dados

Para realizar a aquisição de dados, pensou-se em analisar apenas uma única vaga (Figura 11), visto a capacidade de generalização da rede neural, pois as demais vagas possuem características parecidas.

Para capturar as informações, foi implementado o algoritmo AquisicaoDados.m. Neste algoritmo os dados são capturados pela webcam (Figura 13), ao passo que a imagem é capturada, o usuário informa ao sistema se a mesma representa uma vaga livre (valor zero) ou uma vaga ocupada (valor um). Em seguida, o algoritmo pergunta se uma nova imagem deve ser capturada ou não.

Durante o treinamento, a imagem obtida foi submetida a diferentes padrões de claridade e formas de estacionar o veículo. Apenas 3 veículos foram utilizados neste processo, visando testar posteriormente a capacidade de generalização da rede neural.

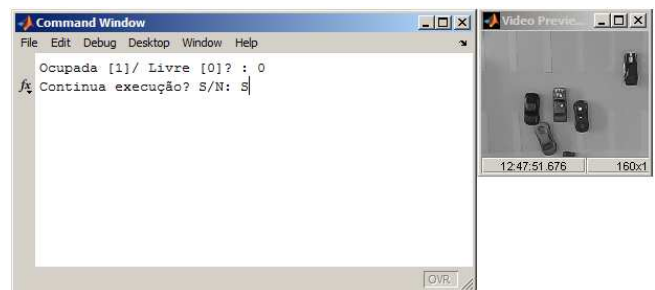


Figura 13 - Aquisição de Dados

C. Treinamento da Rede

Os dados obtidos na etapa anterior foram submetidos à rede para o treinamento, para tal, foi desenvolvido o algoritmo TreinamentoRede.m. Um total de 5 épocas foram necessárias para finalizar o treinamento da rede (Figura 14). Ao todo foram submetidas 63 imagens diferentes, incluindo vagas ocupadas e vazias.

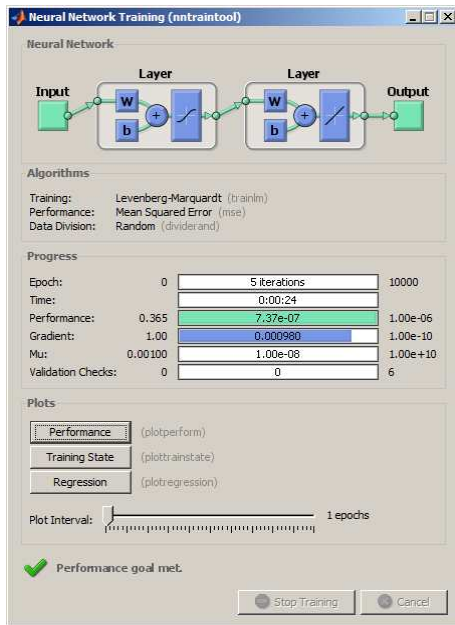


Figura 14 - Treinamento da Rede

Neste ponto, a rede atingiu um estado mínimo de erro entre entrada e saída esperada, com seus pesos atualizados por meio do algoritmo *backpropagation*.

D. Teste de Treinamento

Para validar o treinamento da rede, foi desenvolvido o algoritmo *TesteTreinamento.m*. Neste algoritmo as imagens capturas foram utilizadas para identificar a situação da vaga (ocupada ou vazia). Nesta fase, as imagens capturadas são diferentes daquelas utilizadas no treinamento, visando analisar a generalização da rede.

Do mesmo modo ocorrido no treinamento, diversas situações foram submetidas ao algoritmo, como entrada de veículos não treinados pela rede anteriormente e diferentes posições de estacionamento. É importante frisar que inicialmente um número menor de imagens foram submetidas, obtendo resultados insatisfatórios da rede. Seguindo o processo apresentado na Figura 10, novas imagens foram adquiridas e reapresentadas para um novo treinamento.

A Figura 15 e Figura 16 representam respectivamente a saída da rede quando a vaga está ocupada e livre.

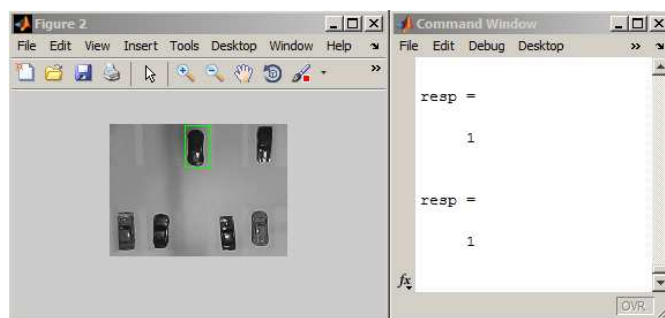


Figura 15 - Teste vaga ocupada

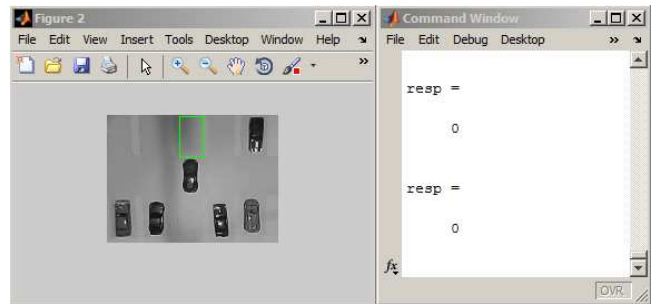


Figura 16 - Teste vaga livre

III. RESULTADOS OBTIDOS

Para apresentar os resultados obtidos pela rede, foram desenvolvidos os algoritmos *ProjetoFinal.m* e *ProjetoFinalRapido.m*. É importante destacar que o segundo algoritmo apresenta apenas um modo mais rápido de apresentar os resultados obtidos pela rede. Enquanto o *ProjetoFinal.m* apresenta o resultado em uma imagem (Figura 17), o *ProjetoFinalRapido.m* apresenta o resultado no formato de uma matriz (Figura 18).

As vagas livres estão representadas na cor verde e as ocupadas na cor vermelha.

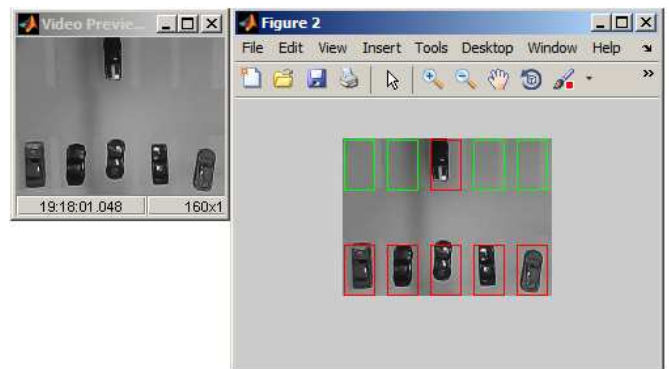


Figura 17 - Projeto Final - Resultado em imagem

Na Figura 18, zeros representam vagas vazias e 1 as vagas ocupadas.

Nestes algoritmos, a rede utilizada para uma única vaga, foi replicada para as demais, partindo do pressuposto de sua capacidade de generalização.

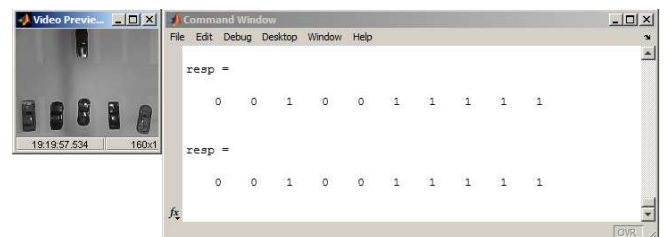


Figura 18 - Projeto Final - Resultado em texto

A Figura 19 apresenta o teste individual de vagas, onde cada vaga foi verificada utilizando apenas um veículo.

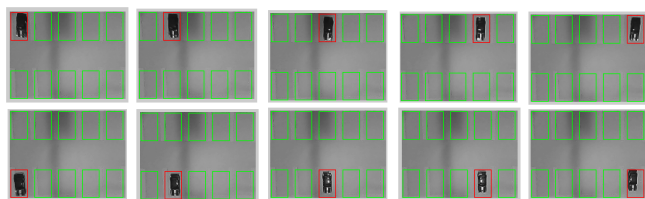


Figura 19 - Teste individual de vagas

A Figura 20 apresenta o teste coletivo de vagas, neste teste a inserção parcial de veículos em algumas vagas (Figura 20c), não representa a ocupação das mesmas, mas sim um momento de locomoção ou manobra do veículo. A Figura 20d mostra que o veículo deverá estar corretamente estacionado para que a vaga seja definida como ocupada.

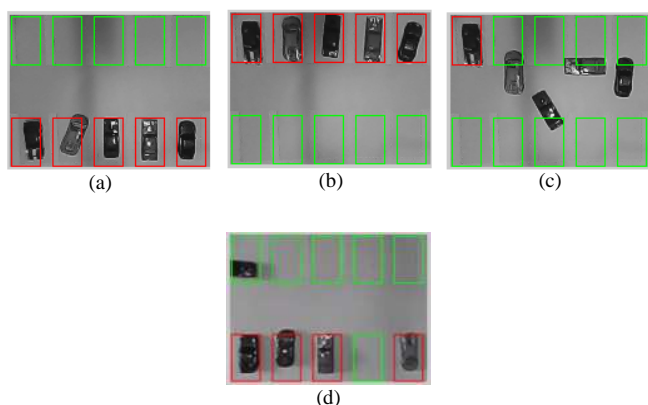


Figura 20 - Teste coletivo de vagas

IV. CONCLUSÃO

Foi notável a viabilidade do uso do protótipo de identificação de vagas em estacionamentos uma vez que os testes realizados mostraram resultados bastante significativos, sendo possível comprovar a eficiência do projeto com o uso de técnicas de processamento de imagem e reconhecimento de padrões a partir de Redes Neurais Artificiais (RNAs) com o auxílio do Matlab.

Para realização deste trabalho, foi considerada a atual necessidade de facilitar o gerenciamento de localização de vagas de trânsito, reduzindo assim o tempo de procura. Para isso, foram observadas as atuais discussões acerca deste referido tema bem como a grande necessidade de soluções mais acessíveis e confiáveis.

Os fatos mais relevantes que se podem levar em consideração é ideia de automatização do sistema, na qual utilizam métodos robustos e confiáveis para uso em estacionamentos abertos onde se tenha boa visão aérea.

Os resultados obtidos criam possibilidades de aperfeiçoamento do projeto para sua implementação, uma vez que um bom treinamento de redes neurais pode viabilizar a adaptação do protótipo para ambientes onde seja possível a captura de imagens. Como o projeto utiliza a captura de imagens em tempo real, esses dados podem ser futuramente

transmitidos para os motoristas via internet, *Bluetooth*, *wi-fi*, dentre outras.

REFERÊNCIAS

- [1] BUARQUE, Ecomensagem, *Asfixia das Cidades*. Disponível em: http://www.ecolnews.com.br/asfixia_das_cidades.htm. Acessado em: 25 de jul. 2011
- [2] MOURA, Elizângela. Zona Azul deve começar a funcionar em setembro, no centro de Mossoró. *Correio Mossoró*. Rio Grande do Norte, 28 jul. 2011. Disponível em: http://www.correiodatarde.com.br/editorias/correio_mossoro-64228. Acesso em: 28 jul. 2011.
- [3] AZEVEDO, Gabriel. Falta de estacionamento vira problema na praia. *Gazeta do povo*. Paraná, 20/01/2011. Disponível em: <http://www.gazetadopovo.com.br/verao/conteudo.phtml?id=1088408>. Acesso em: 28 jul. 2011.
- [4] MADER Helena; TEMÓTEO Antonio. Vagas subterrâneas devem ser solução na área central de Brasília. *Correio Braziliense*. Brasília 15/07/2011. Disponível em: http://www.correiobraziliense.com.br/app/noticia/cidades/2011/07/15/interior_cidadesdf,261323/vagas-subterraneas-devem-ser-solucao-na-area-central-de-brasilia.shtml. Acessado em: 28 jul. 2011.
- [5] JACOB, Lupa, Zona Azul. *Diário de Ciclista*. Pará 14/05/2010. Disponível em: <http://diariodeciclista.blogspot.com/2010/05/zona-azul.html>. Acessado em: 25 de jul. 2011
- [6] NORVING, Stuart Russell Peter, *Inteligência Artificial*. 2 Edição. Editora Campus: Rio de Janeiro, 2004.
- [7] BRAGA, Antônio de Pádua; LUDERMIR, Teresa Bernarda; CARVALHO, André Ponce de Leon F. de Carvalho. *Redes Neurais Artificiais: teoria e aplicações*. Editora LTC: Rio de Janeiro, 2000.
- [8] MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. *Processamento Digital de Imagens*, Rio de Janeiro: Brasport, 1999. ISBN 8574520098.
- [9] MARIA, L. G. F. *Processamento Digital de Imagens*. INPE, Junho de 2000.
- [10] SILVA, Renato Ramos da. *Reconhecimento de imagens digitais utilizando redes neurais artificiais*. 2005. 51 f. Monografia (Graduação em Ciência da Computação). Universidade Federal de Lavras, Lavras – MG.
- [11] GONZALEZ, R. C. WOODS, R. E. *Processamento de Imagens Digitais*. EDITORA EDGARD BLÜCHER LTDA: São Paulo, 2000.
- [12] BRAGA, Antônio de Pádua; LUDERMIR, Teresa Bernarda; CARVALHO, André C. P. de Leon Ferreira. *Redes Neurais Artificiais: teoria e aplicações*. 2 edição. Editora LTC: Rio de Janeiro, 2007.
- [13] HAYKIN, S. *Redes Neurais: Princípios e prática*. McMaster University, Hamilton, Ontário, Canadá, 2001.
- [14] BITTENCOURT, J. R. *Sistemas Inteligentes baseados em Redes Neurais Artificiais aplicados ao Processamento de Imagens*, In: I Workshop de inteligência artificial, 2000, apostila-seminário, Santa Cruz do Sul, UNISC – Universidade de Santa Cruz do Sul - Departamento de Informática, 2000.
- [15] CARLOS, J. F. P. *Aplicação de Redes Neurais no Processamento Digital de Imagens*. Universidade Federal de Minas Gerais, 1994.
- [16] MEDEIROS, L. F. de. *Redes neurais em delphi*. Florianópolis: Visual Books Editora, 2003, 115p.