



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Daniel Felipe Silva Santos

Reconhecimento de Veículos em Imagens Coloridas Utilizando
Máquinas de Boltzmann Profundas e Projeção Bilinear

São José do Rio Preto
2017

Daniel Felipe Silva Santos

Reconhecimento de Veículos em Imagens Coloridas Utilizando
Máquinas de Boltzmann Profundas e Projeção Bilinear

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Orientador: Prof. Dr. Aparecido Nilceu Marana

São José do Rio Preto
2017

Santos, Daniel Felipe Silva.

Reconhecimento de veículos em imagens coloridas utilizando Máquinas de Boltzmann Profundas e Projeção Bilinear / Daniel Felipe Silva Santos. -- São José do Rio Preto, 2017

61 f. : il.

Orientador: Aparecido Nilceu Marana

Dissertação (mestrado) – Universidade Estadual Paulista "Júlio de Mesquita Filho", Instituto de Biociências, Letras e Ciências Exatas

1. Computação. 2. Processamento de imagens. 3. Análise de imagens – Processamento de dados. 4. Veículos. 5. Projeção bilinear. I. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas. II. Título.

CDU – 518.72:76

Ficha catalográfica elaborada pela Biblioteca do IBILCE
UNESP - Campus de São José do Rio Preto


Daniel Felipe Silva Santos

Reconhecimento de Veículos em Imagens Coloridas Utilizando
Máquinas de Boltzmann Profundas e Projeção Bilinear

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

 Comissão Examinadora

Prof. Dr. Aparecido Nilceu Marana
UNESP – Câmpus de Bauru
Orientador


Prof. Dr. João Paulo Papa
UNESP – Câmpus de Bauru


Prof. Dr. Marcelo Andrade da Costa Vieira
EESC/USP

São José do Rio Preto
14 de agosto de 2017

AGRADECIMENTOS

Aos meus pais, por todo suporte nas horas mais difíceis;

Ao meu orientador Professor Dr. Aparecido Nilceu Marana por todo apoio e paciência;

À Professora Dra. Bárbara Stolte Bezerra por todo o incentivo;

Aos pesquisadores Geoffrey Hinton e Ruslan Salakhutdinov por compartilharem com o mundo suas inúmeras pesquisas e ideias inovadoras na área de Aprendizado de Máquinas;

Finalmente, agradeço aos meus grandes amigos Rafael e Gustavo, que tive a honra de conhecer ao longo desta minha jornada, pelas inúmeras horas que passamos pesquisando e aprendendo uns com os outros.

RESUMO

Neste trabalho é proposto um método para reconhecer veículos em imagens coloridas baseado em uma rede neural *Perceptron* Multicamadas pré-treinada por meio de técnicas de aprendizado em profundidade, sendo uma das técnicas composta por Máquinas de Boltzmann Profundas e projeção bilinear e a outra composta por Máquinas de Boltzmann Profundas Multinomiais e projeção bilinear. A proposição deste método justifica-se pela demanda cada vez maior da área de Sistemas de Transporte Inteligentes. Para se obter um reconhecedor de veículos robusto, a proposta é utilizar o método de treinamento inferencial não-supervisionado Divergência por Contraste em conjunto com o método inferencial Campos Intermediários, para treinar múltiplas instâncias das redes profundas. Na fase de pré-treinamento local do método proposto são utilizadas projeções bilineares para reduzir o número de nós nas camadas da rede. A junção das estruturas em redes profundas treinadas separadamente forma a arquitetura final da rede neural, que passa por uma etapa de pré-treinamento global por Campos Intermediários. Na última etapa de treinamentos a rede neural *Perceptron* Multicamadas (MLP) é inicializada com os parâmetros pré-treinados globalmente e a partir deste ponto, inicia-se um processo de treinamento supervisionado utilizando gradiente conjugado de segunda ordem. O método proposto foi avaliado sobre a base BIT-Vehicle de imagens frontais de veículos coletadas de um ambiente de tráfego real. Os melhores resultados obtidos pelo método proposto utilizando rede profunda multinomial foram de 81,83% de acurácia média na versão aumentada da base original e 91,10% na versão aumentada da base combinada (Carros, Caminhões e Ônibus). Para a abordagem de redes profundas não multinomiais os melhores resultados foram de 81,42% na versão aumentada da base original e 91,13% na versão aumentada da base combinada. Com a aplicação da projeção bilinear, houve um decréscimo considerável nos tempos de treinamento das redes profundas multinomial e não multinomial, sendo que no melhor caso o tempo de execução do método proposto foi 5,5 vezes menor em comparação com os tempos das redes profundas sem aplicação de projeção bilinear.

Palavras-chave: Reconhecimento de Veículos. Máquinas de Boltzmann Profundas. Máquinas de Boltzmann Profundas Multinomiais. Projeção Bilinear.

ABSTRACT

In this work it is proposed a vehicle recognition method for color images based on a Multilayer Perceptron neural network pre-trained through deep learning techniques (one technique composed by Deep Boltzmann Machines and bilinear projections and the other composed by Multinomial Deep Boltzmann Machines and bilinear projections). This proposition is justified by the increasing demand in Traffic Engineering area for the class of Intelligent Transportation Systems. In order to create a robust vehicle recognizer, the proposal is to use the inferential unsupervised training method of Contrastive Divergence together with the Mean Field inferential method, for training multiple instances of deep models. In the local pre-training phase of the proposed method, bilinear projections are used to reduce the number of nodes of the neural network. The combination of the separated trained deep models constitutes the final recognizer's architecture, that yet will be global pre-trained through Mean Field. In the last phase of training the Multilayer Perceptron neural network is initialized with globally pre-trained parameters and from this point, a process of supervised training starts using second order conjugate gradient. The proposed method was evaluated over the BIT-Vehicle database of frontal images of vehicles collected from a real road traffic environment. The best results obtained by the proposed method that used multinomial deep models were 81.83% of mean accuracy in the augmented original database version and 91.10% in the augmented combined database version (Cars, Trucks and Buses). For the non-multinomial deep models approach, the best results were 81.42% in the augmented version of the original database and 91.13% in the augmented version of the combined database. It was also observed a significant decreasing in the training times of the multinomial deep models and non-multinomial deep models with bilinear projection application, where in the best case scenario the execution time of the proposed method was 5.5 times lower than the deep models that did not use bilinear projection.

Keywords: *Vehicle Recognition. Deep Boltzmann Machines. Multinomial Deep Boltzmann Machines. Bilinear Projection.*

Lista de ilustrações

Figura 1 – Modelos BM e RBM	19
Figura 2 – Amostragem Gibbs	21
Figura 3 – Arquitetura de rede DBM de duas camadas	22
Figura 4 – Representação de interações por MF	23
Figura 5 – Modelo 2D-DBN Reconhecedor de Veículos	29
Figura 6 – Processo de Particionamento de Imagem em Blocos	32
Figura 7 – Filtros dos Blocos da Posição Inicial	32
Figura 8 – Fluxograma de Pré-Treinamento Local.	36
Figura 9 – Projeção de imagem RGB e inicialização de GRBM	37
Figura 10 – Inicialização de DBM global	39
Figura 11 – Inicialização de MLP	40
Figura 12 – Base de Dados BIT-Vehicle	43
Figura 13 – Normalização de imagens de veículos	46
Figura 14 – Aumento artificial de Minivans	47

Lista de tabelas

Tabela 1 – Acurácias médias - Grupo de Avaliação de Classes Não Combinadas . .	51
Tabela 2 – Tempos médios de treinamento - Grupo de Avaliação de Classes Não Combinadas	51
Tabela 3 – Acurácias médias - Grupo de Avaliação de Classes Combinadas	52
Tabela 4 – Tempos médios de treinamento - Grupo de Avaliação de Classes Com- binadas	53

Lista de abreviaturas e siglas

BIT	<i>Beijing Institute of Technology</i>
BLAS	<i>Basic Linear Algebra Subprograms</i>
BM	<i>Boltzmann Machine</i>
CD	<i>Contrastive Divergence</i>
CIFAR	<i>Canadian Institute For Advanced Research</i>
CLAHE	<i>Contrast Limited Adaptive Histogram Equalization</i>
CNN	<i>Convolutional Neural Network</i>
CUDA	<i>Compute Unified Device Architecture</i>
DBM	<i>Deep Boltzmann Machine</i>
DBN	<i>Deep Belief Network</i>
GPU	<i>Graphics Processing Unit</i>
GRBM	<i>Gaussian Restricted Boltzmann Machine</i>
HSV	<i>Hue Saturation Value</i>
ITS	<i>Intelligent Transportation System</i>
LPP	<i>Locality Preserving Projections</i>
MCMC	<i>Markov Chain Monte Carlo</i>
MDBM	<i>Multinomial Deep Boltzmann Machine</i>
MF	<i>Mean Field</i>
MFA	<i>Marginal Fisher Analysis</i>
MLP	<i>Multilayer Perceptron</i>
OpenCV	<i>Open Source Computer Vision Library</i>

RBM	<i>Restricted Boltzmann Machine</i>
RGB	<i>Red Green Blue</i>
2D-LDA	<i>Bilinear Discriminant Analysis</i>
2D-DBM	<i>Bilinear Deep Boltzmann Machine</i>
2D-DBN	<i>Bilinear Deep Belief Network</i>
2D-MDBM	<i>Bilinear Multinomial Deep Boltzmann Machine</i>

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos	14
1.2	Organização da Dissertação	14
2	APRENDIZADO EM PROFUNDIDADE	16
2.1	Modelagem Estatística por Grafos Direcionados	16
2.2	Modelagem Estatística por Grafos Não Direcionados	17
2.2.1	Amostragem de Unidades	17
2.2.2	Máquinas de Boltzmann	19
2.2.3	Máquinas de Boltzmann Restritas	20
2.2.4	Máquinas de Boltzmann Profundas	22
2.2.5	Máquinas de Boltzmann Profundas Multinomiais	25
3	PROJEÇÃO BILINEAR	26
4	TRABALHOS CORRELATOS	28
4.1	Deteccção de Veículos Utilizando Redes de Crença em Profundidade Bilineares	28
4.2	Aprendizado em Profundidade Aplicado em Blocos de Imagens Coloridas	31
5	MÉTODO PROPOSTO	34
5.1	Pré-Treinamento Local	35
5.2	Pré-Treinamento Global	37
5.3	Treinamento e Utilização do Classificador MLP	39
6	MATERIAL E METODOLOGIA DE AVALIAÇÃO	42
6.1	Recursos Computacionais	42
6.2	Base de Dados BIT-Vehicle	42
6.3	Metodologia de Avaliação do Método Proposto	44
6.3.1	Grupo de Avaliação de Classes Não Combinadas	44

6.3.1.1	Teste com Base Original	45
6.3.1.2	Teste com Base Normalizada	45
6.3.1.3	Teste com Base Aumentada	46
6.3.2	Grupo de Avaliação de Classes Combinadas	48
7	RESULTADOS E DISCUSSÃO	49
7.1	Resultados do Grupo de Avaliação de Classes Não Combinadas . . .	50
7.2	Resultados do Grupo de Avaliação de Classes Combinadas	52
7.3	Resultados Obtidos com CNN	53
8	CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS . .	55
8.1	Contribuições deste Trabalho	56
8.2	Propostas para Trabalhos Futuros	56
8.3	Trabalhos Publicados e Submetidos	56
	REFERÊNCIAS	58

1 Introdução

Seguindo uma tendência mundial de avanço tecnológico, diversos setores têm passado por um processo gradativo de reestruturação. Dentre estes setores encontra-se o setor de transportes, para o qual muitos sistemas computacionais, incluídos na categoria de ITS (*Intelligent Transportation System*), vêm sendo desenvolvidos. Exemplos de sistemas projetados com a finalidade de auxiliar motoristas na prevenção de colisões no trânsito são mostrados nos trabalhos de Birrell, Fowkes e Jennings (2014), Baek e Kim (2014), Teoh e Bräunl (2012) e Almagambetov, S. e Casares (2015), enquanto que exemplos de sistemas desenvolvidos para tarefas de monitoramento envolvendo extração de métricas para análise do fluxo de tráfego podem ser encontrados nos trabalhos de Lv et al. (2015), Lu et al. (2014) e Song, Song e Wang (2014).

Os ITSs são sistemas que, em geral, são compostos por diversos módulos, tais como módulos de detecção, rastreamento e reconhecimento de veículos como mostra Salvi (2014). O módulo de reconhecimento, que em geral atua nas últimas etapas de execução do ITS, tem um impacto considerável nos processos de tomada de decisão executados pelo sistema, como por exemplo classificar um veículo detectado na imagem em uma classe pré-determinada. O objetivo deste trabalho é desenvolver um método robusto e eficiente para classificação de veículos por meio da aplicação de técnicas de pré-treinamento de rede neural utilizando aprendizado em profundidade e projeção bilinear, motivado pelos trabalhos de Sun et al. (2014), Hu et al. (2014) e Dong et al. (2015), que obtiveram sucesso nesta tarefa.

De acordo com Hinton, Osindero e Teh (2006) e Salakhutdinov e Hinton (2012) as principais vantagens de se utilizar aprendizado em profundidade são: (i) a capacidade de extração automática de características por meio de modelagem da estrutura estatística intrínseca à imagem analisada; (ii) a capacidade de atuar em problemas de ordem discriminativa e também de generalização; (iii) a capacidade de gerar parâmetros ótimos¹ de inicialização de redes neurais *Perceptron* Multicamadas (do inglês, *Multilayer Perceptron - MLP*).

¹ A inicialização ótima tenta evitar que as buscas pelo ponto de mínimo global da função de erro da rede neural fiquem “presas” em pontos de mínimo locais, o que geralmente acontece com mais frequência em técnicas de inicialização por randomização de valores.

Em contrapartida às vantagens apresentadas, existem duas principais desvantagens associadas aos métodos de treinamento utilizando aprendizado em profundidade. A primeira desvantagem diz respeito ao esforço computacional despendido pela realização de operações matriciais, pela utilização de técnicas de amostragem e pela execução de métodos de otimização de segunda ordem, como a técnica gradiente conjugado de LeCun et al. (2012) e Navon e Legler (1987).

A segunda desvantagem é a dificuldade para definir a configuração da arquitetura e dos parâmetros de treinamento dos modelos. No caso da arquitetura, devem ser definidas as quantidades de camadas intermediárias e as quantidades de neurônios que estas camadas irão conter. Com relação aos parâmetros existe uma série de valores de taxas de aprendizado, quantidades de interações e tamanho de lotes, que se indevidamente configurados podem ocasionar sérios problemas de convergência dos métodos de treinamento. Zhong, Yan e Yang (2011) e Wang, Cai e Chen (2014) utilizam projeções bilineares para definir a arquitetura de uma rede neural MLP e inicializar seus parâmetros mais adequadamente em uma etapa anterior ao pré-treinamento. Krizhevsky (2009) e Hu et al. (2014) propõem como solução para definir a arquitetura (também de uma rede neural MLP) a aplicação de uma estratégia de pré-treinamento utilizando subdivisão de imagem colorida em regiões de tamanho 8×8 .

Neste trabalho também são oferecidas soluções para estas desvantagens, sendo que para superar a desvantagem de configuração de arquitetura são utilizados os métodos Máquinas de Boltzmann Profundas (do inglês, *Deep Boltzmann Machine* - DBM) de Salakhutdinov e Hinton (2012), e Máquinas de Boltzmann Profundas Multinomiais (do inglês, *Multinomial Deep Boltzmann Machine* - MDBM) de Salakhutdinov, Tenenbaum e Torralba (2013), além da utilização do método de projeção bilinear 2D-LDA (*Bilinear Discriminant Analysis*) de Yang et al. (2005), capaz de reduzir a quantidade de neurônios das camadas escondidas preservando a acurácia da rede neural MLP.

Para minimizar ainda mais a desvantagem de demora no tempo de treinamento são empregadas técnicas de paralelização de código via GPU (*Graphics Processing Unit*) por meio de programação em CUDA (*Compute Unified Device Architecture*).

1.1 Objetivos

O objetivo deste trabalho é desenvolver um método robusto e eficiente para reconhecimento de veículos, por meio de suas classificações em classes pré-definidas, utilizando técnicas de pré-treinamento de rede neural que combinam, em uma primeira abordagem, Máquinas de Boltzmann Profundas e projeções bilineares e, em uma segunda abordagem, Máquinas de Boltzmann Profundas Multinomiais e projeções bilineares.

1.2 Organização da Dissertação

Esta dissertação de mestrado contém os seguintes capítulos:

Capítulo 1: Apresenta uma breve introdução sobre a importância do reconhecimento de veículos, principalmente para sistemas ITS, apresenta as perspectivas de aplicação de aprendizado em profundidade para criação de classificadores de veículo mais eficientes e robustos e apresenta os objetivos desta dissertação de mestrado;

Capítulo 2: Apresenta os fundamentos básicos necessários para compreender os métodos de aprendizado de Máquinas de Boltzmann Profundas e Máquinas de Boltzmann Profundas Multinomiais, utilizados neste trabalho;

Capítulo 3: Apresenta de forma concisa a fundamentação teórica necessária para compreender a técnica de projeção bilinear utilizada neste trabalho;

Capítulo 4: Apresenta dois trabalhos principais utilizados como base para o desenvolvimento desta dissertação de mestrado;

Capítulo 5: Apresenta o modo de construção e treinamento do classificador proposto nesta dissertação de mestrado para o reconhecimento de veículos;

Capítulo 6: Apresenta os principais recursos de *hardware*, *softwares* e base de dados utilizados para o desenvolvimento deste trabalho, além da metodologia de avaliação empregada;

Capítulo 7: Apresenta e discute os resultados obtidos pelo classificador proposto, com as diferentes abordagens de aprendizado em profundidade que foram utilizadas neste trabalho;

Capítulo 8: Apresenta as conclusões desta dissertação de mestrado mediante a análise dos resultados obtidos, propõe alguns trabalhos futuros e lista as publicações realizadas.

2 Aprendizado em Profundidade

Segundo Bengio (2009), diante de uma imagem de entrada, um modelo de redes neurais em profundidade ideal visa simular a capacidade do sistema visual e cognitivo humano de capturar dados e extrair, de forma incremental, características relevantes destes mesmos dados. Indo mais além no processo de extração de informação, as redes ideais seriam capazes de gerar modelos representativos cada vez mais complexos até chegar em níveis de representação de contexto, nível semântico. No aprendizado em profundidade o modo de extração de informações é essencialmente baseado na modelagem estatística por grafos, que combina a teoria de grafos (DIESTEL, 2005) com a teoria estatística (MOORE; MCCABE; CRAIG, 2009) tornando possível representar de forma compacta e eficiente a estrutura estatística da imagem analisada.

Nas Seções 2.1 e 2.2 são apresentados os fundamentos teóricos relativos respectivamente à modelagem estatística por grafos direcionados e à modelagem estatística por grafos não direcionados.

2.1 Modelagem Estatística por Grafos Direcionados

Segundo Wainwright e Jordan (2008), nos modelos por grafos direcionados podem ser destacados dois componentes principais. O primeiro deles é o sentido de orientação das arestas que conectam os vértices, componente utilizado para expressar o relacionamento existente entre os vértices. Neste caso o vértice de onde parte a aresta é considerado o vértice pai, enquanto que o vértice de chegada é considerado o filho e o segundo componente a ser mencionados é o modelo de distribuição de probabilidades condicional gerado implicitamente por meio deste relacionamento, como mostra a Equação (1),

$$P_d(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n) = \prod_{\mathbf{v}_i \in \mathbf{v}} P_i(\mathbf{v}_i | \mathbf{h}_j), \quad (1)$$

sendo que $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ representa o conjunto filho e $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m\}$ representa o conjunto pai.

2.2 Modelagem Estatística por Grafos Não Direcionados

Na modelagem estatística por grafos não direcionados não existe uma relação direta condicional, relação vértice pai e vértice filho exibida nas modelagens estatísticas por grafos direcionados. Wainwright e Jordan (2008) mostram que este tipo de modelagem leva em consideração o relacionamento entre as cliques¹ geradas pelas interações simétricas entre os vértices do grafo em questão, como mostrado na Figura 1(b) pelas conexões entre os conjuntos de vértices \mathbf{v} e \mathbf{h} . Desta forma, tomando por base uma clique A e uma clique B , considerando-se aqui que $A \not\subset B$, é possível substituir a relação de condicionalidade por uma relação de compatibilidade entre os conjuntos A e B . Neste caso, a compatibilidade ficará expressa pelo produto cartesiano calculado entre os vetores aleatórios de A e de B , lembrando que participam destes conjuntos, além dos vértices, as variáveis aleatórias a eles associados. Como consequência é possível derivar desta abordagem uma nova forma fatorada para o modelo probabilístico, denotada pela Equação (2)

$$P_{nd}(v_1, v_2, \dots, v_n) = \frac{1}{Z} \prod_{C \in S} \psi_C(v_C), \quad (2)$$

onde S representa o conjunto das cliques máximas, Z denota uma função partição (soma de todas as possibilidades existentes de comparação entre as cliques máximas utilizada para normalizar P_{nd}) e $\psi_C(v_C)$ denota a função que mede a compatibilidade entre as cliques formadas pelo conjunto v_C .

2.2.1 Amostragem de Unidades

Ao tratar os nós dos modelos de rede em grafos apresentados ao longo da Subseção 2.2 como neurônios, surge a necessidade de associar a estes elementos funções de ativação, cuja finalidade é simular o comportamento biológico dos gatilhos de impulso, do inglês *spike trigger averages*, atuantes no processo de cognição humana. Estes gatilhos são acionados quando o valor de potencial máximo de ação de um neurônio é atingido. Após o disparo, o sinal é propagado de um neurônio a outro e o processo então se repete. Para mais informações sobre este processo vide (DAYAN; ABBOTT, 2005).

¹ De acordo com Luce e Perry (1949) cliques são subconjuntos contidos em grafos e que são compostos por três ou mais vértices simetricamente conectados entre si.

A função de ativação geralmente utilizada nos modelos de redes neurais é a função sigmóide, apresentada na Seção 2.2.4, devido principalmente ao seu comportamento não linear, seu conjunto imagem estar em uma faixa de valores reais com variação entre 0 e 1, mesma faixa de valores de probabilidade, e também devido à sua caracterização como uma função diferenciável em todo o seu domínio.

A amostragem faz parte do processo de simulação computacional do estado de um nó com base no seu valor de ativação. Os dois principais tipos de amostragem são a Bernoulli e a Gaussiana, que associam-se respectivamente com o valor de ativação do nó gerado por uma função sigmóide e pelo valor do nó gerado por uma função linear.

Amostragem Bernoulli: Nesta técnica de amostragem os eventos $\{X = x\}$ onde $x \in \{0, 1\}$, ou seja, a probabilidade de uma variável aleatória alterar seu estado será dada por $P(X = 0)$ ou por $P(X = 1)$ em acordo com uma distribuição Bernoulli, neste caso discreta, de probabilidades. Na etapa de propagação do sinal de ativação da camada visível \mathbf{v} para a camada escondida \mathbf{h} , a amostragem é calculada sobre a distribuição à posteriori da camada escondida e pode ser descrita pela função sigmóide na forma da Equação (3),

$$P(h_j = 1|\mathbf{v}) = \frac{1}{1 + \exp(-b_j - \sum_i v_i w_{ij})}. \quad (3)$$

onde b_j indica o viés de um nó j da camada \mathbf{h} , $v_i \in \mathbf{v}$ e w_{ij} indica o calor da aresta que conecta o nó j da camada \mathbf{h} ao nó i da camada \mathbf{v} do modelo em grafos. O mesmo vale para a propagação de sinal na direção oposta, sentido \mathbf{h} para \mathbf{v} , com sua representação dada pela Equação (4),

$$P(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-c_i - \sum_j h_j w_{ij})}, \quad (4)$$

onde c_i indica o viés de um nó i da camada \mathbf{v} .

Amostragem Gaussiana: Neste método de amostragem utilizam-se unidades visíveis lineares com adição independente de ruído gaussiano. Desta forma a

Equação (4) passa a ser representada por,

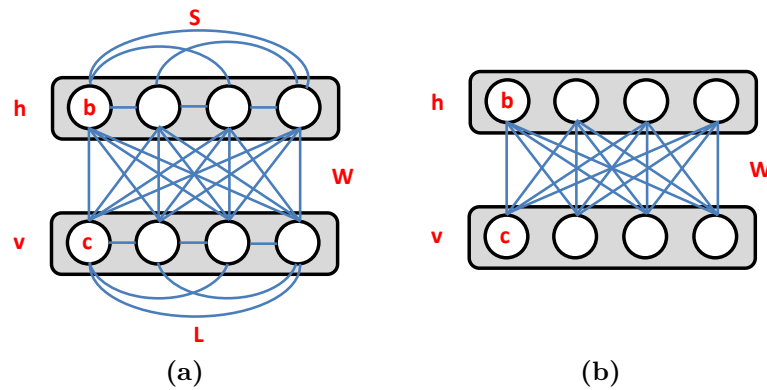
$$p(v_i|\mathbf{h}) = \mathcal{N}\left(v_i \left| \sum_j h_j w_{ij} + c_i, \sigma_i^2 \right.\right), \quad (5)$$

onde uma distribuição normal gaussiana $\mathcal{N}(\cdot)$ de média $\sum_j h_j w_{ij} + c_i$ e variância σ_i^2 é calculada para cada $v_i \in \mathbf{v}$.

2.2.2 Máquinas de Boltzmann

Segundo Rumelhart, McClelland e Group (1986) Máquinas de Boltzmann (do inglês, *Boltzmann Machine* - BM), são grafos não-direcionados formados por unidades estocásticas binárias simetricamente conectadas entre si, como mostra a Figura 1(a), onde **S** indica os valores das arestas que conectam os nós da camada **h** entre si, **L** indica os valores das conexões feitas entre os nós da camada **v** e **W** indica os valores das conexões efetuadas entre as camadas **v** e **h**.

Figura 1 – Modelos de Redes. (a) Máquina de Boltzmann (BM);
(b) Máquina de Boltzmann Restrita (RBM).



Fonte: Elaborada pelo autor.

Ainda de acordo com Rumelhart, McClelland e Group (1986) as BMs são também modelos de energia, que por sua vez, vide Equação (6), são governadas por uma distribuição de Boltzmann que em equilíbrio térmico é descrita por:

$$P(\mathbf{v}, \mathbf{h}; \Theta) = \frac{e^{-E(\mathbf{v}, \mathbf{h})}}{Z(\Theta)}, \quad (6)$$

em que \mathbf{v} e \mathbf{h} denotam as camadas visível e escondida respectivamente da BM, $\Theta = \{\mathbf{L}, \mathbf{S}, \mathbf{W}\}$ faz referência ao conjunto de parâmetros da BM, $Z(\Theta)$ diz respeito à função partição:

$$Z(\Theta) = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}, \quad (7)$$

e $E(\mathbf{v}, \mathbf{h})$ descreve a função de energia do modelo dada por:

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i < j} v_i v_j \mathbf{L}_{ij} - \sum_{k < l} h_k h_l \mathbf{S}_{kl} - \sum_{i, k} v_i h_k \mathbf{W}_{ik}. \quad (8)$$

Em uma primeira tentativa para encontrar $\Theta = \{\mathbf{L}, \mathbf{S}, \mathbf{W}\}$ capaz de minimizar a energia $E(\mathbf{v}, \mathbf{h})$ da Equação (8), Rumelhart, McClelland e Group (1986) propõem a utilização de um algoritmo denominado *annealing* simulado. Técnica esta baseada no cálculo da razão entre os valores das probabilidades de ativação dos nós da BM, como mostra a Equação (9).

$$\frac{P_\alpha}{P_\beta} = e^{-(E_\alpha - E_\beta)/T}. \quad (9)$$

A razão calculada pela Equação (9) remete a estimativa das diferenças entre as energias de duas configurações globais diferentes, denominadas α e β , ao ser alterada a temperatura T do modelo. Altas temperaturas fazem com que o algoritmo não fique preso à mínimos locais, enquanto que baixas temperaturas fazem com que um possível bom ponto de mínimo local não seja ignorado durante o processo de busca.

2.2.3 Máquinas de Boltzmann Restritas

As Máquinas de Boltzmann Restritas (do inglês, *Restricted Boltzmann Machine* - RBM) surgem ao ser imposta sobre as BMs uma restrição que elimina as conexões entre os nós pertencentes à uma mesma camada, como mostra a Figura 1(b). Apesar desta modificação alterar a capacidade de generalização do modelo em grafos, ela gera um grande impacto na redução da complexidade do modelo de energia quando comparado com a BM,

sendo que as inferências passam a ser feitas de forma direta. A Equação (10) mostra a função de energia da RBM,

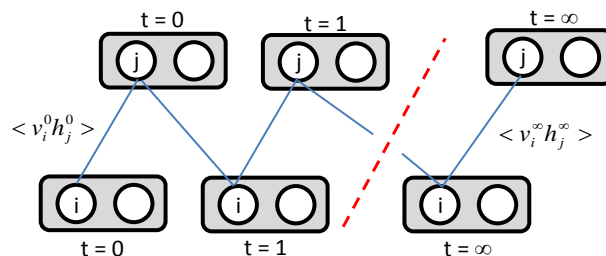
$$E(\mathbf{v}, \mathbf{h}) = - \sum_i v_i c_i - \sum_k h_k b_k - \sum_{i,k} v_i h_k \mathbf{W}_{ik}. \quad (10)$$

Para minimizar a energia da RBM Carreira-Perpiñán e Hinton (2005) desenvolvem o método Divergência de Contraste (do inglês, *Contrastive Divergence* - CD), cujo principal objetivo é encontrar o conjunto Θ de parâmetros que maximiza a Equação (11) de verossimilhança logarítmica que modela estatisticamente a camada visível \mathbf{v} da RBM de uma forma mais eficiente que o método *annealing* simulado apresentado na Seção 2.2.2.

$$\frac{\partial \log P(\mathbf{v})}{\partial \Theta} = \left\langle \frac{\partial E(\mathbf{x}; \Theta)}{\partial \Theta} \right\rangle_0 - \left\langle \frac{\partial E(\mathbf{x}; \Theta)}{\partial \Theta} \right\rangle_\infty \quad (11)$$

A otimização por CD mede a diferença entre duas correlações médias calculadas entre os nós da camada visível e os nós da camada escondida, representado na Equação (11) por $\mathbf{x} = \mathbf{v}\mathbf{h}$, de maneira independente. Como mostra a Figura 2, o CD é um processo Markoviano que parte de um estado observável em um instante $t = 0$, onde \mathbf{v} pertence ao conjunto de dados observáveis, e é executado por um período de tempo suficientemente grande para que o sistema atinja o estado de equilíbrio, instante representado na Figura 2 por ∞ .

Figura 2 – Amostragem Gibbs.

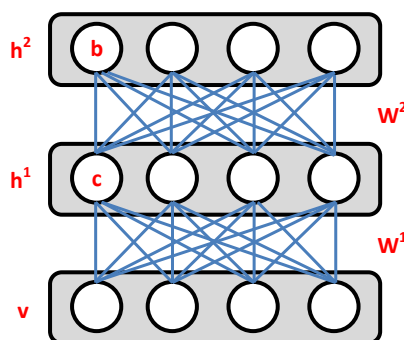


Fonte: Hinton, Osindero e Teh (2006, p. 1534).

2.2.4 Máquinas de Boltzmann Profundas

De acordo com Salakhutdinov e Hinton (2012) os modelos DBMs (Deep Boltzmann Machines), apresentam pelo menos três vantagens, sendo elas: a habilidade em aprender internamente representações dos dados de entrada, onde quanto mais afastada da camada de entrada estiver a camada escondida, maior será sua capacidade de capturar estruturas estatísticas complexas; a eficiência no processo de cálculo das ativações dos neurônios, que é aumentada devido à estrutura de conexões herdada do modelo RBM; a capacidade de utilização de *feedback* na extração de características de mais alto nível, que por sua vez são utilizadas para resolver incertezas à respeito de características extraídas por níveis intermediários. A Figura 3 ilustra uma arquitetura DBM de duas camadas.

Figura 3 – Arquitetura de rede DBM de duas camadas.



Fonte: Elaborada pelo autor.

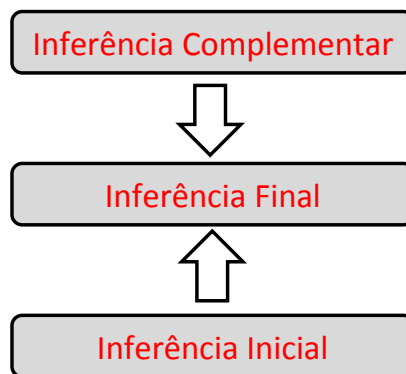
Segundo Salakhutdinov e Hinton (2012) o treinamento de DBMs pode ser subdividido em dois estágios, sendo eles o estágio de pré-treinamento em gride e o estágio de pré-treinamento por Campos Médios (do inglês, *Mean Field* - MF), (PETERSON; ANDERSON, 1987) e (WELLING; HINTON, 2002). O pré-treinamento em gride consiste no treinamento de uma primeira RBM, aplicando o processo de treinamento por CD², como apresentado na Subseção 2.2.3, no posicionamento da mesma na base da estrutura DBM, na utilização dos valores amostrados, vide Equação (3), da camada escondida da primeira RBM como valores de entrada de uma segunda RBM a ser treinada também por CD. O processo é repetido para cada camada escondida adicionada à DBM.

² Salakhutdinov e Hinton (2012) propõe multiplicar por 2 os parâmetros que conectam camadas intermediárias durante os processos de inferência e reconstrução por causa da falta de *feedback* inicial das camadas.

O estágio de pré-treinamento por MF tem como principal objetivo aproximar a verdadeira distribuição à posteriori $P(\mathbf{B}|\mathbf{v}; \Theta)$ calculada sobre as camadas escondidas, $\mathbf{B} = \{\mathbf{h}^1, \mathbf{h}^2, \dots, \mathbf{h}^l\}$ sendo l a quantidade máxima de camadas da rede, de uma distribuição unimodal fatorável $Q^{MF}(\mathbf{B}|\mathbf{v}; \boldsymbol{\mu})$, onde o parâmetro Θ permanece fixo enquanto o conjunto $\boldsymbol{\mu}$ é determinado por um processo de interações MF.

A Figura 4 mostra uma representação simplificada do processo de interações MF em DBMs. Neste caso é possível observar que a ativação da camada intermediária Inferência Final é composta em parte pelo sinal de *feedback* gerado pela camada Inferência Complementar e pela estimativa feita pela camada de mais baixo nível Inferência Inicial.

Figura 4 – Representação de interações por Mean Field (MF).



Fonte: Elaborada pelo autor.

Tomando como exemplo a rede DBM de duas camadas da Figura 3, é possível aproximar a distribuição de probabilidades à posteriori $P(\mathbf{B}|\mathbf{v}; \Theta)$ calculada sobre conjunto $\mathbf{B} = \{\mathbf{h}^1, \mathbf{h}^2\}$ de camadas escondidas por meio da técnica de inferência variacional MF utilizando-se para isto a distribuição:

$$Q^{MF}(\mathbf{B}|\mathbf{v}; \boldsymbol{\mu}) = \prod_{n=1}^2 \left[\prod_{k=1}^{F_n} q(h_k^n) \right]. \quad (12)$$

onde F_n representa a quantidade total de nós da camada escondida n e $q(h_k^n = 1) = \mu$, vide Equação (3).

A solução para a Equação (12) está justamente em encontrar o conjunto de parâmetros $\boldsymbol{\mu} = \{\mu_1, \mu_2\}$ dependente do conjunto de dados observáveis \mathbf{v} . Isto é feito

por meio da resolução de equações denominadas fixas, como mostram Salakhutdinov e Larochelle (2014), que podem apresentar três formas diferentes de acordo com as camadas que participam da interação MF. Denotando $\text{sigm}(\bar{x}) = 1/(1 + e^{-\bar{x}})$, a primeira forma é mostrada pela Equação (13),

$$\mu_j^1 \leftarrow \text{sigm} \left(\sum_{i=1}^D W_{ij}^1 v_i + \sum_{k=1}^{F_2} W_{jk}^2 \mu_k^2 \right), \quad (13)$$

que representa a interação entre a camada visível e a primeira camada escondida, a forma dois é exibida pela Equação (14),

$$\mu_k^2 \leftarrow \text{sigm} \left(\sum_{j=1}^{F_1} W_{jk}^2 \mu_j^1 + \sum_{m=1}^{F_3} W_{km}^3 \mu_m^3 \right), \quad (14)$$

que representa as interações entre duas camadas escondidas intermediárias e por fim a forma três é apresentada pela Equação (15)

$$\mu_m^3 \leftarrow \text{sigm} \left(\sum_{k=1}^{F_2} W_{km}^3 \mu_k^2 \right), \quad (15)$$

representando as interações entre as duas últimas camadas escondidas de uma rede DBM. Neste exemplo de duas camadas apenas a Equação (13) e a Equação (15) seriam utilizadas. Ao final do processo de aplicação da aproximação por MF à rede DBM, Salakhutdinov e Hinton (2012) assumem que os parâmetros \mathbf{v} e \mathbf{B}^{MF} são suficientes para representar o modelo estatístico dependente de dado observável, termo positivo da Equação (11), tendo em vista que a rede encontrar-se-á em equilíbrio térmico.

Para encontrar o modelo estatístico dependente do dado não observável, termo negativo da Equação (11), é utilizado um processo de amostragem *Gibbs* persistente. De acordo com Tieleman (2008) após inicializar o algoritmo de PCD (*Persistent Contrastive Divergence*) sobre um conjunto de M partículas fantasia³ $X^t = \{\tilde{x}^{t,1}, \dots, \tilde{x}^{t,M}\}$ randomicamente inicializadas em um instante t a partir de uma distribuição uniforme $D_{uni} = [0, 1]$, será utilizado um operador de transição $T_{\theta_t}(\tilde{x}_{t+1} \leftarrow \tilde{x}_t)$ por certa quantidade de vezes até que o ponto de equilíbrio térmico seja alcançado. Essencialmente isto é o mesmo que aplicar

³ As partículas fantasia correspondem ao conjunto de entrada cujos valores são inicializado randomicamente.

a técnica CD no instante t utilizando como ponto de partida do processo de amostragem *Gibbs*, vide Figura 2, os dados gerados pela aplicação da mesma técnica CD no instante $t - 1$. Desde que a quantidade de interações do algoritmo PCD seja mantida baixa e a taxa de aprendizado sofra decaimento, como sugerido por Salakhutdinov e Hinton (2012), o processo de aproximação estocástica por simulação MCMC (*Markov Chain Monte Carlo*) será capaz de encontrar o conjunto ótimo de parâmetros $\Theta = \{\mathbf{W}^{(1,\dots,l)}, \mathbf{b}^{(1,\dots,l)}, \mathbf{c}^{(1,\dots,l)}\}$ para a rede DBM.

2.2.5 Máquinas de Boltzmann Profundas Multinomiais

As Máquinas de Boltzmann Profundas Multinomiais (do inglês, *Multinomial Deep Boltzmann Machines* - MDBM) foram criadas por Salakhutdinov, Tenenbaum e Torralba (2013) com o intuito de aumentar a capacidade do modelo DBM em aprender, por meio das camadas escondidas de mais alto nível, estruturas estatísticas mais complexas a partir dos dados de entrada da rede. A arquitetura de uma MDBM não difere muito da arquitetura de uma DBM, exceto pelo acréscimo de uma camada *softmax* no topo da pilha de RBMs onde a Equação (3) do cálculo das ativações dos neurônios é modificado por:

$$P(h_k^\gamma | \mathbf{h}^{\gamma-1}) = \frac{\exp(\sum_l w_{lk}^k h_l^2)}{\sum_{n=1}^k \exp(\sum_l w_{ln}^k h_l^2)}, \quad (16)$$

onde k indica o k -ésimo nó da camada *softmax* e γ indica a posição que a camada ocupa na hierarquia de camadas escondidas.

Segundo Salakhutdinov, Tenenbaum e Torralba (2013) as atividades da camada h^γ são modeladas por uma distribuição condicional multinomial onde as k unidades *softmax* compartilham o mesmo conjunto das l conexões que partem da camada escondida $h^{\gamma-1}$. Este tipo de configuração faz com que todos os neurônios da camada h^γ funcionem como um único neurônio amostrado várias vezes.

3 Projeção Bilinear

Segundo Yang et al. (2005), Ye, Janardan e Li (2005) e Liang, Li e Shi (2008), dado um conjunto contendo k matrizes $A = \{X^1, X^2, X^3, \dots, X^k\}$ onde $X^k \in \mathbb{R}^{m \times n}$ em que \bar{A} indica a matriz média e \bar{A}^c indica a matriz média calculada sobre todos os elementos pertencentes a classe c , o tipo específico de projeção bilinear 2D-LDA (*Bilinear Discriminant Analysis*), tem por objetivo produzir uma representação mais compacta para os elementos de A . Desta forma após a aplicação da projeção bilinear sobre o conjunto A ele passará a ser representado por $D \in \mathbb{R}^{p \times q}$ onde $p \times q < m \times n$. As matrizes que irão projetar A em D , representadas aqui genericamente por ϕ , precisam satisfazer a função de custo,

$$J(\phi) = \frac{\phi^T G_b \phi}{\phi^T G_w \phi}. \quad (17)$$

A Equação (17) mostra que as matrizes ótimas de projeção serão aquelas capazes de maximizar as distâncias entre os elementos de A que pertencerem à classes diferentes e minimizar as distâncias entre os elementos que pertencerem à uma mesma classe. O processo de otimização consistirá, partindo da matriz de projeção das colunas de A , $\phi = \mathbf{U}$, em encontrar as matrizes de separação da Equação (17) onde $G_b = S_b$ para a matriz de separação inter classe e $G_w = S_w$ para a matriz de separação intra classe onde,

$$S_b = \frac{1}{k} \sum_{i=1}^c K_i (\bar{A}^i - \bar{A})^T (\bar{A}^i - \bar{A}) \quad (18)$$

e

$$S_w = \frac{1}{k} \sum_{i=1}^c \sum_{j=1}^{K_i} (A_j^i - \bar{A}^i)^T (A_j^i - \bar{A}^i). \quad (19)$$

Observando que em (18) e (19) K denota o conjunto de números inteiros positivos que indicam a quantidade máxima de elementos em cada classe e também que A_j^i indica o j -ésimo elemento de A pertencente à classe i .

De acordo com (YANG et al., 2005) como a matriz de projeção \mathbf{U} deve ser constituída por vetores de características não correlacionados a geração de \mathbf{U} consiste na resolução da decomposição,

$$S_w^{-1}S_b = \lambda_j u_j, \quad (20)$$

onde $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_q$ constitui o conjunto de autovalores associados com o conjunto de autovetores $\{u_1, u_2, \dots, u_q\}$ que irão compor a matriz de projeção $\mathbf{U} \in \mathbb{R}^{n \times q}$.

Após encontrar $B = A\mathbf{U}$ o próximo passo da otimização será encontrar $\phi = \mathbf{V}$ seguindo basicamente a mesma abordagem já utilizada para encontrar \mathbf{U} . Nesta etapa o processo para encontrar as matrizes de separação $G_b = H_b$ and $G_w = H_w$ passa a ser descrito por,

$$H_b = \frac{1}{k} \sum_{i=1}^c K_i (\bar{B}^i - \bar{B})(\bar{B}^i - \bar{B})^T \quad (21)$$

e

$$H_w = \frac{1}{k} \sum_{i=1}^c \sum_{j=1}^{K_i} (B_j^i - \bar{B}^i)(B_j^i - \bar{B}^i)^T. \quad (22)$$

Assim como \mathbf{U} a matriz de projeção das linhas de A é encontrada resolvendo-se a decomposição,

$$H_w^{-1}H_b = \epsilon_j v_j, \quad (23)$$

sendo que $\mathbf{V} \in \mathbb{R}^{m \times p}$ e $\epsilon_1 \geq \epsilon_2 \geq \epsilon_3 \geq \dots \geq \epsilon_p$ forma o conjunto de autovalores associados com o conjunto de autovetores $\{v_1, v_2, \dots, v_p\}$.

O processo de otimização é interrompido quando (17) alcança um valor limite. Ao final da otimização a solução para a projeção bilinear de A será dada pelo conjunto $D = \mathbf{V}^T A\mathbf{U}$.

4 Trabalhos Correlatos

Neste capítulo são apresentados os principais aspectos do trabalho de Wang, Cai e Chen (2014) e de Krizhevsky (2009), selecionados por condensarem praticamente toda a informação necessária para compreensão do método proposto nesta dissertação de mestrado. Wang, Cai e Chen (2014) apresentam uma solução que utiliza projeções bilineares combinadas com Redes de Crença em Profundidade (do inglês, *Deep Belief Networks* - DBN) para gerar um eficiente reconhecedor de veículo. Krizhevsky (2009) apresenta uma técnica de pré-treinamento que utiliza múltiplas instâncias de modelos DBNs geradas a partir de partes extraídas de imagens coloridas.

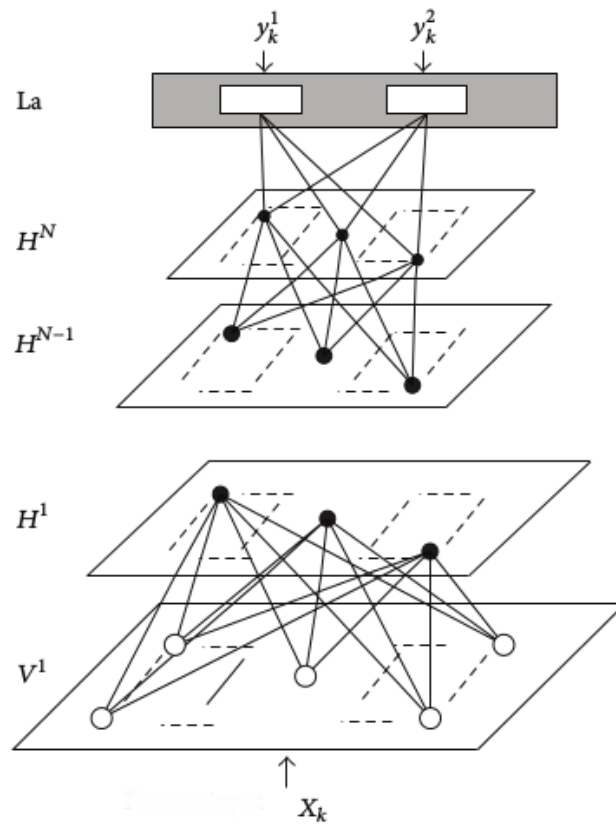
4.1 Detecção de Veículos Utilizando Redes de Crença em Profundidade Bilineares

Wang, Cai e Chen (2014) propõem um método de detecção de veículos utilizando uma combinação de técnicas de Aprendizado em Profundidade com Projeções Bilineares. A técnica de projeção é aplicada sobre a camada de entrada e sobre as camadas escondidas, o que permite inicializar os parâmetros do modelo de Redes de Crença Profundas Bilineares (do inglês, *Bilinear Deep Belief Networks* - 2D-DBN). A rede bilinear é pré-treinada de forma não-supervisionada por meio do empilhamento de Máquinas de Boltzmann Restritas, treinadas separadamente utilizando-se Divergência de Contraste, e de forma supervisionada aplicando-se *backpropagation*. A Figura 5 mostra o modelo da rede 2D-DBN utilizado.

Inicialmente, um conjunto X com imagens positivas e negativas, neste caso referentes à veículos e não veículos respectivamente, e um conjunto Y de rótulos são construídos. Desta forma tem-se que,

$$\begin{aligned} X &= [X_1, X_1, \dots, X_k], \\ Y &= [y_1, y_2, \dots, y_k]. \end{aligned} \tag{24}$$

Considerando que X_k possui m linhas e n colunas e que y_k assumirá apenas os valores $(1, 0)$ e $(0, 1)$ para indicar que a imagem k é ou não um veículo. Para automatizar

Figura 5 – Modelo 2D-DBN Reconhecedor de Veículos.

Fonte: (WANG; CAI; CHEN, 2014, p. 3).

o processo de construção das camadas intermediárias da rede são utilizadas projeções bilineares. Em linhas gerais pode-se dizer que este processo busca projetar apenas a informação mais discriminativa da camada inferior para a superior, como mostra a Figura 5. Esta é uma técnica baseada em LPP (*Locality Preserving Projections*) e MFA (*Marginal Fisher Analysis*), abordadas respectivamente em (HE; NIYOGI, 2004) e (YAN et al., 2005). A função de otimização que busca encontrar as matrizes de projeção $U \in R^{m \times p}$ e $G \in R^{n \times q}$ é dada pela Equação (25),

$$\operatorname{argmax}_{U, G} J(U, G) = \sum_{i, j} \|U^T(\mathbf{X}_i - \mathbf{X}_j)G\|^2(\alpha B_{ij} - (1 - \alpha)W_{ij}), \quad (25)$$

em que as distâncias interclasses, \mathbf{B} , e intra-classes, \mathbf{W} , são também consideradas e

calculadas por meio de:

$$B_{ij} = \begin{cases} \frac{1}{n_d} - \frac{1}{n_c}, & \text{se } y_i^c = y_j^c = 1 \\ \frac{1}{n_d}, & \text{caso contrário,} \end{cases} \quad (26)$$

$$W_{ij} = \begin{cases} \frac{1}{n_c}, & \text{se } y_i^c = y_j^c = 1 \\ 0, & \text{caso contrário,} \end{cases}$$

onde n_d indica a quantidade de amostras ao serem contabilizadas todas as classes e n_c denota a quantidade de amostras considerando-se apenas a classe c .

A Equação (25) é então maximizada em detrimento da minimização das distâncias intra-classes e da maximização das distâncias interclasses. Pode-se observar que para elementos da mesma classe o termo $E_{ij} = \alpha B_{ij} - (1 - \alpha)W_{ij}$ será negativo, levando a Equação (25) a minimizar as distâncias. Por outro lado, considerando a comparação entre elementos de classes distintas, o termo E_{ij} será positivo, o que gera um efeito de maximização entre as distâncias.

Wang, Cai e Chen (2014) utilizam a mesma técnica empregada por Zhong, Yan e Yang (2011) para encontrar \mathbf{U} e \mathbf{G} . A técnica consiste em fixar \mathbf{U} ou \mathbf{G} , encontrar os k autovalores do polinômio característico dado por $\mathbf{D}_F \mathbf{L} = \lambda \mathbf{L}$, onde \mathbf{F} indica o termo fixo e \mathbf{L} o termo livre e onde a matriz das distâncias é dada pela Equação (27)

$$\mathbf{D}_F = \sum_{ij} E_{ij} (X_i - X_j) \mathbf{F} \mathbf{F}^T (X_i - X_j)^T. \quad (27)$$

Segundo Zhong, Yan e Yang (2011), a eliminação dos autovetores de \mathbf{D}_F associados aos autovalores negativos garante a maximização da Equação (25).

Após terem sido encontradas as matrizes \mathbf{U} e \mathbf{G} ótimas, o processo evolui para a segunda etapa. Nesta segunda etapa a informação projetada será utilizada para inicializar a primeira RBM, sendo esta composta pela camada de entrada V^1 e camada escondida H^1 , como mostra a Figura 5. Todo o processo de extração de características passa a ser em função da otimização do conjunto $\Psi = \{A^1, c^1, b^1, \dots, A^z, c^z, b^z\}$ sendo que z denota a quantidade de RBMs a serem empilhadas, onde $A^z = U^z \otimes G^z$ demonstra que a z -ésima RBM será inicializada pelas z -ésimas matrizes \mathbf{U} e \mathbf{G} de projeção. Os termos c e b são os vieses associados respectivamente à camada visível e escondida de cada RBM.

A última fase de treinamento, a etapa de refino de parâmetros, consiste em aplicar um treinamento supervisionado à rede em função do conjunto de rótulos \mathbf{Y} . No trabalho os autores utilizam o algoritmo de *backpropagation* para realizar esta última etapa.

4.2 Aprendizado em Profundidade Aplicado em Blocos de Imagens Coloridas

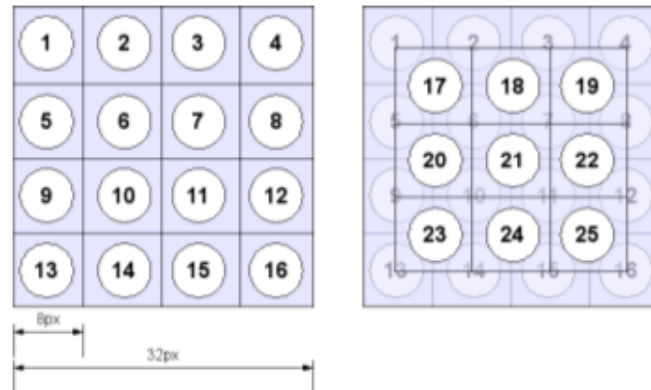
Krizhevsky (2009) desenvolve um método de treinamento baseado na utilização de aprendizado em profundidade para resolver o problema de reconhecimento de objetos em imagens coloridas, o qual é avaliado por meio da base CIFAR-10 (HINTON; KRIZHEVSKY; NAIR, 2009), que compreende um total de 60000 imagens de dimensões 32×32 pertencentes ao espaço RGB (*Red Green Blue*) de cores. Dentre estas imagens, 50000 são utilizadas na etapa de treinamento e 10000 são utilizadas na etapa de teste.

Esta base apresenta dois grandes desafios, sendo eles a quantidade de classes a serem corretamente classificadas, neste caso 10, e a interferência que pode ser gerada pelas diferentes tonalidades dos cenários de fundo onde se inserem os objetos, como por exemplo interferências por camuflagens. Na tentativa de superar estes problemas e obter uma boa taxa de classificação, Krizhevsky (2009) sugere particionar as imagens e utilizar várias DBNs locais, sendo uma para cada partição. Isto é sugerido para melhorar o processo de extração de características e conseqüentemente melhorar o resultado obtido pelo treinamento não-supervisionado e posteriormente pelo treinamento supervisionado. A Figura 6 ilustra o processo de particionamento da imagem de entrada em 25 blocos.

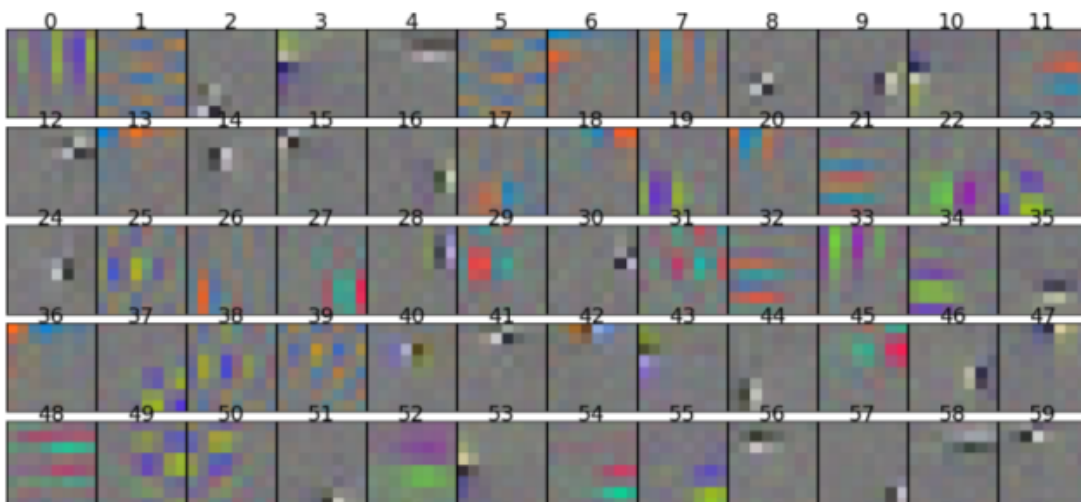
As etapas que compõem todo o processo de treinamento são basicamente três. Na primeira delas as imagens são particionadas em regiões de tamanhos 8×8 , como mostra a Figura 6. Cada região contendo 50000 imagens é conectada à uma camada escondida contendo 300 nós, formando assim várias RBMs. Cada RBM é submetida à um processo de treinamento não-supervisionado utilizando Divergência de Contraste de uma interação (CD-1).

Além dos 25 blocos presentes na Figura 6, Krizhevsky (2009) define o vigésimo sexto bloco composto por uma versão reduzida da imagem de tamanho 32×32 para 8×8 . A Figura 7 mostra exemplos de filtros gerados pela RBM associada ao bloco 1.

Na segunda etapa as RBMs são horizontalmente conectadas, ou seja, a RBM

Figura 6 – Processo de Particionamento de Imagem em Blocos

Fonte: (KRIZHEVSKY, 2009, p. 20).

Figura 7 – Exemplo de filtros gerados a partir da região 1 das imagens de treinamento da base CIFAR-10.

Fonte: (KRIZHEVSKY, 2009, p. 21).

associada ao bloco 1, por exemplo, quando conectada com a RBM do bloco 2 dará origem a uma RBM composta por 600 nós na camada escondida e 384 nós na camada visível, e de acordo com Krizhevsky (2009) as conexões entre nós visíveis e nós escondidos pertencentes à blocos diferentes devem ser inicializadas com o valor 0, exceto no caso do vigésimo sexto bloco que tem os valores das conexões entre seus nós visíveis e todos os outros nós escondidos inicializados pela média do seu valor original. Exemplificando, a conexão entre o nó um da camada visível do vigésimo sexto bloco com o nó um da camada escondida do

primeiro bloco será inicializado pelo valor da conexão entre o nó um da camada visível do vigésimo sexto bloco com o nó um da camada escondida deste mesmo bloco ponderado por 26. Krizhevsky (2009) também sugere que o valor do viés visível c da estrutura combinada seja a média calculada entre os termos de todas as RBMs concatenadas. O processo pode ser facilmente replicado para uma Rede de Crença em Profundidade (do inglês, *Deep Belief Network* - DBN) ao ser aplicado um processo de empilhamento nas RBMs treinadas bloco a bloco.

Na terceira etapa a nova estrutura RBM passa por um novo processo de treinamento não-supervisionado e ao final por um treinamento supervisionado utilizando *backpropagation*.

5 Método Proposto

Este trabalho propõe um método eficiente de pré-treinamento de uma rede neural MLP de duas camadas¹ utilizada na classificação de veículos. O método proposto utiliza duas abordagens distintas, sendo que na primeira delas são utilizadas como técnicas de geração de parâmetros ótimos de inicialização da rede neural MLP o treinamento de Máquinas de Boltzmann Profundas (do inglês, *Deep Boltzmann Machines* - DBM), vide nas Subseções 2.2.4 e 2.2.1, e a técnica de projeção bilinear 2D-LDA (*Bilinear Discriminant Analysis*) descrita no Capítulo 3.

Na segunda abordagem são utilizadas como técnicas ótimas de inicialização da rede neural MLP o treinamento de Máquinas de Boltzmann Profundas Multinomiais (do inglês, *Multinomial Deep Boltzmann Machines* - MDBM), vide Subseção 2.2.5, e também a técnica de projeção bilinear 2D-LDA. Para destacar em uma só nomenclatura os pontos principais de cada abordagem, optou-se pela utilização das siglas 2D-DBM (*Bilinear Deep Boltzmann Machine*) como referência ao classificador MLP inicializado por meio de Máquina de Boltzmann Profunda Bilinear e 2D-MDBM (*Bilinear Multinomial Deep Boltzmann Machine*) como referência ao classificador MLP inicializado por meio de Máquina de Boltzmann Profunda Multinomial Bilinear.

As estratégias de treinamento local e treinamento global utilizadas em ambas abordagens foram baseadas em grande parte nos trabalhos apresentados nas Seções 4.1 e 4.2, com a diferença de que naqueles trabalhos os autores utilizaram uma Rede de Crença em Profundidade (DBN) para realizar o pré-treinamento em profundidade e no caso desta dissertação são utilizadas Máquinas de Boltzmann Profundas (DBMs) e Máquinas de Boltzmann Profundas Multinomiais (MDBMs) com o intuito de melhorar o processo automático de extração de características de imagens, tendo em vista que os modelos DBMs são capazes de encontrar estruturas estatísticas complexas nos dados de treinamento, vide Subseção 2.2.4. O método de pré-treinamento de classificador MLP de veículos proposto neste trabalho diferencia-se também do método apresentado na Seção 4.1 com relação à forma de aplicação da técnica de projeção bilinear, aqui adaptada para imagens coloridas com o intuito de adicionar mais informação relevante ao processo de projeção bilinear.

¹ Neste caso foram contabilizadas apenas as camadas intermediárias da rede neural, excluindo-se portanto do cálculo a camada de entrada e a de saída.

Para facilitar o entendimento do método de treinamento da rede MLP proposto, a sequência de treinamentos foi decomposta em três etapas: pré-treinamento local, descrito na Seção 5.1; pré-treinamento global, descrito na Seção 5.2; e etapa de treinamento supervisionado, descrita na Seção 5.3, que resulta na geração do classificador de veículos 2D-DBM para a abordagem de pré-treinamento com DBM e do classificador 2D-MDBM no caso da abordagem de pré-treinamento por DBM Multinomial.

5.1 Pré-Treinamento Local

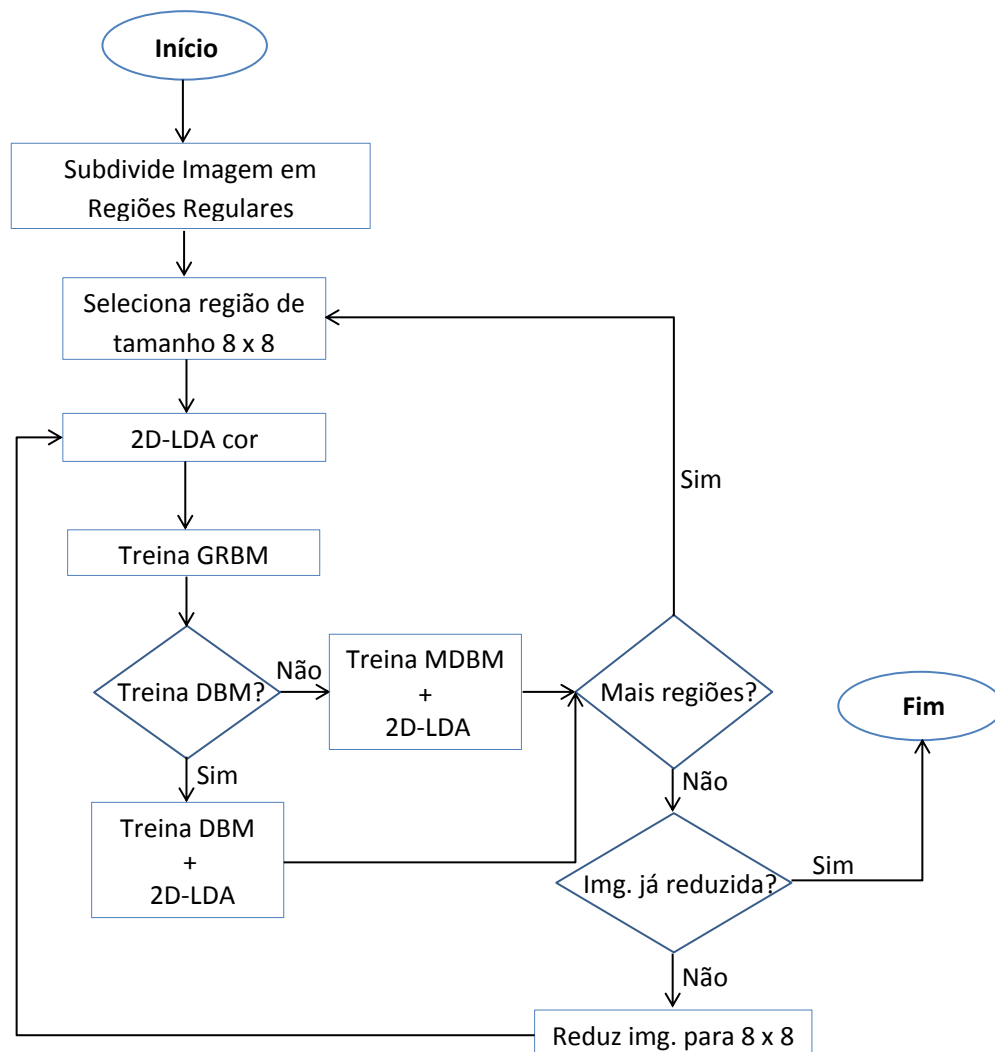
A etapa de pré-treinamento local treina a abordagem de rede profunda escolhida (neste caso DBM ou MDBM) para cada bloco extraído da imagem subdividida, conforme ilustrado na Figura 6, e desta forma é capaz de melhorar a extração de características da estrutura em redes profundas utilizada, Seção 5.2. No fluxograma apresentado na Figura 8 observa-se que em um primeiro momento é realizada a subdivisão das imagens de tamanho 32×32 originando os 25 blocos de imagens de tamanho 8×8 que são ilustrados na Figura 6.

Em um segundo momento as imagens particionadas são projetadas durante a etapa 2D-LDA cor. Esta etapa é ilustrada pela Figura 9 por Projeção Bilinear de Imagem Colorida e visa calcular os três pares de matrizes de projeção bilinear, sendo o primeiro par definido por $\mathbf{U}^{8 \times q^{(R)}}$ e $\mathbf{V}^{8 \times p^{(R)}}$; o segundo por $\mathbf{U}^{8 \times q^{(G)}}$ e $\mathbf{V}^{8 \times p^{(G)}}$; e o terceiro por $\mathbf{U}^{8 \times q^{(B)}}$ e $\mathbf{V}^{8 \times p^{(B)}}$, onde R denota à projeção de canal Vermelho, G a projeção do canal Verde e B a projeção do canal Azul.

Como ilustrado pela Figura 9 após terem sido calculadas as matrizes de projeção inicia-se o processo de construção de uma Máquina Restrita de Boltzmann Gaussiana² (do inglês, *Gaussian Restricted Boltzmann Machine* - GRBM) onde as linhas em azul indicam valores pertencentes à $W^{(f)}$, sendo que $W^{(f)} = \mathbf{V}^{(f)T} \otimes \mathbf{U}^{(f)}$ para $f \in \{R, G, B\}$ e as linhas tracejadas em preto indicam valores inicializados em 0^3 . Na etapa seguinte a GRBM é treinada e sua camada \mathbf{h} é utilizada como entrada da rede profunda (DBM ou MDBM) de pré-treinamento. Como indica o fluxograma da Figura 8 o processo de

² De acordo com Nair e Hinton (2009) para que redes profundas sejam capazes de aprender modelos de distribuição gaussianos é necessário aproximar estes modelos à distribuições binomiais. Em outras palavras a camada \mathbf{h} da GRBM treinada será uma boa aproximação binomial de \mathbf{v} para a abordagem de rede profunda escolhida.

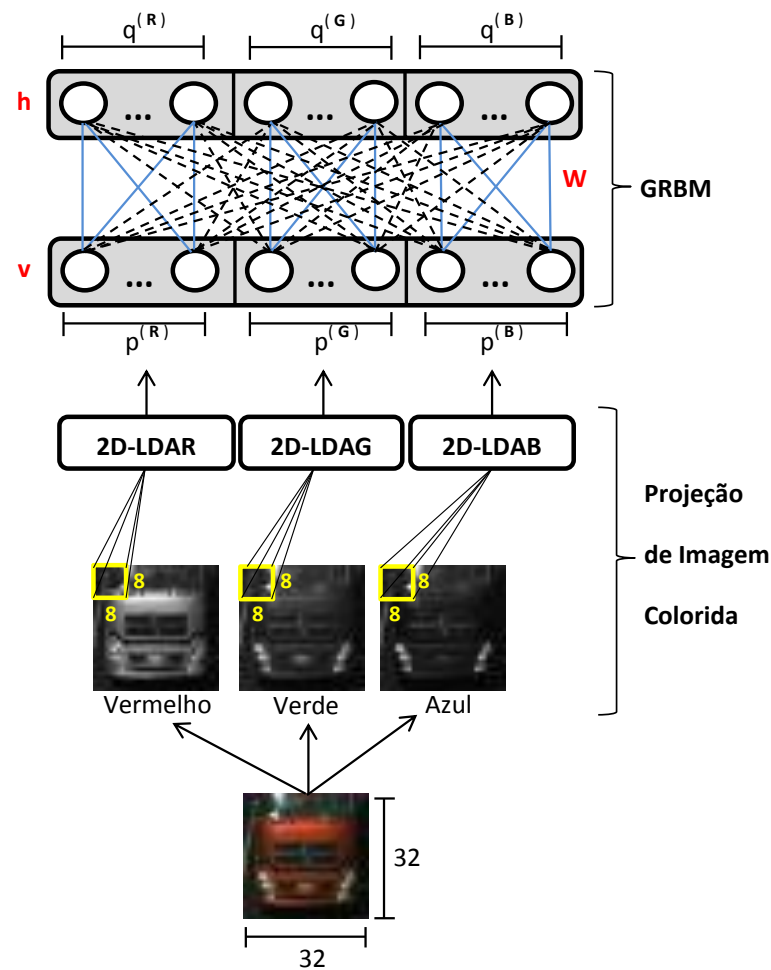
³ A justificativa para a inicialização com o valor 0 pode ser encontrada na Seção 4.2.

Figura 8 – Fluxograma de Pré-Treinamento Local.

Fonte: Elaborada pelo autor.

pré-treinamento da rede profunda já engloba a projeção bilinear, neste caso a aplicação da técnica 2D-LDA torna-se mais simples, tendo em vista que a GRBM da região 8×8 possui uma única camada \mathbf{h} diferentemente da imagem de entrada de três canais.

Após este primeiro treinamento da rede profunda (DBM ou MDBM) o fluxo de pré-treinamento local que vai da etapa “Seleciona região de tamanho 8×8 ” até a etapa “Treina GRBM” é executado para cada uma das 24 regiões restantes, como ilustrado pela Figura 6. Como mostra o fluxograma da Figura 8 ao terem sido processadas todas as 25 regiões, o último teste verifica se o vigésimo sexto bloco precisa ser gerado e caso ainda haja essa necessidade, então de acordo com o fluxograma da Figura 8 a imagem originalmente

Figura 9 – Projeção de imagem RGB e Inicialização de GRBM.

Fonte: Elaborada pelo autor.

no tamanho 32×32 é reduzida para o tamanho 8×8 e o processo de pré-treinamento local de rede profunda é realizado para esta nova imagem, observando que a utilização da imagem original reduzida como uma nova região a ser processada segue a abordagem explicada na Seção 4.2. O pré-treinamento local é encerrado logo após a vigésima sexta região ter sido processada. Como subproduto da etapa de pré-treinamento local serão geradas 26 redes profundas.

5.2 Pré-Treinamento Global

Neste segundo estágio de aplicação de pré-treinamentos as 26 estruturas em redes profundas geradas a partir da etapa de pré-treinamento local da Seção 5.1 são combinadas

para formar uma única estrutura de rede profunda (DBM ou MDBM de acordo com a abordagem escolhida). Desta forma os parâmetros da rede profunda são inicializados por $\Theta = \{\mathbf{W}_1^l, \dots, \mathbf{W}_{26}^l, \mathbf{c}_1^l, \dots, \mathbf{c}_{26}^l, \mathbf{b}_1^l, \dots, \mathbf{b}_{26}^l\}$, onde l indica qual a posição do parâmetro na pilha de RBMs.

A inicialização dos vieses de generalização e reconhecimento, \mathbf{c} e \mathbf{b} respectivamente, é feita por meio da concatenação em forma de um único vetor coluna⁴ de todos os vetores coluna menores associados à cada rede profunda já treinada localmente. Esta estratégia foi adotada porque os vetores coluna menores raramente possuem as mesmas dimensões, dificultando a utilização de outras técnicas para agregar estes elementos. No caso das dimensões serem as mesmas Krizhevsky (2009) sugere como forma de agregação o cálculo de vetor médio das 26 estruturas em redes profundas locais e posterior concatenação das 25 réplicas deste vetor médio.

A Figura 10 ilustra mais detalhadamente o processo de inicialização⁵ do parâmetro \mathbf{W} da estrutura de rede profunda, que neste caso é muito parecido com a processo de inicialização do parâmetro \mathbf{W} da GRBM ilustrado pela Figura 9. A principal diferença encontra-se na concatenação da vigésima sexta estrutura de rede profunda localmente treinada na Seção 5.1, que neste caso específico segue o método de concatenação explicado na Seção 4.2.

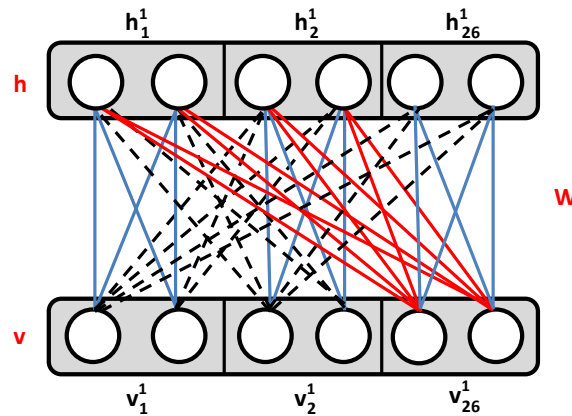
As linhas azuis na Figura 10 dizem respeito aos parâmetros das estruturas de rede profundas geradas pelo pré-treinamento local da Seção 5.1, as linhas pretas tracejadas indicam a inicialização de valores em 0.0 e as linhas vermelhas representam o método de inicialização específico utilizado apenas na concatenação com a vigésima sexta estrutura de rede profunda.

Tendo sido feita a inicialização da estrutura de rede profunda, são aplicadas as sequências de treinamento por inferência variacional (treinamento em gride e treinamento por MF como explicado na Seção 2.2.4).

⁴ Os vieses associados à cada nó visível serão armazenados no vetor coluna \mathbf{c} , enquanto que os vieses associados à cada nó escondido serão armazenados no vetor coluna \mathbf{b} .

⁵ Como o processo de inicialização da estrutura de rede profunda é igual em todos os seus níveis, optou-se por ilustrar apenas o processo aplicado no primeiro nível.

Figura 10 – Inicialização de DBM global.



Fonte: Elaborada pelo autor.

5.3 Treinamento e Utilização do Classificador MLP

Na fase de treinamento do classificador a rede neural MLP primeiramente é inicializada com os parâmetros \mathbf{W} e \mathbf{b} gerados pelo pré-treinamento global, como ilustrado pela Figura 11.

Após inicialização a rede neural é treinada por um método de gradiente conjugado de segunda ordem que utiliza busca linear e método de avaliação de direção conjugada Polack-Ribiere (mais informações podem ser encontradas nos trabalhos de LeCun et al. (2012) e de Navon e Legler (1987)).

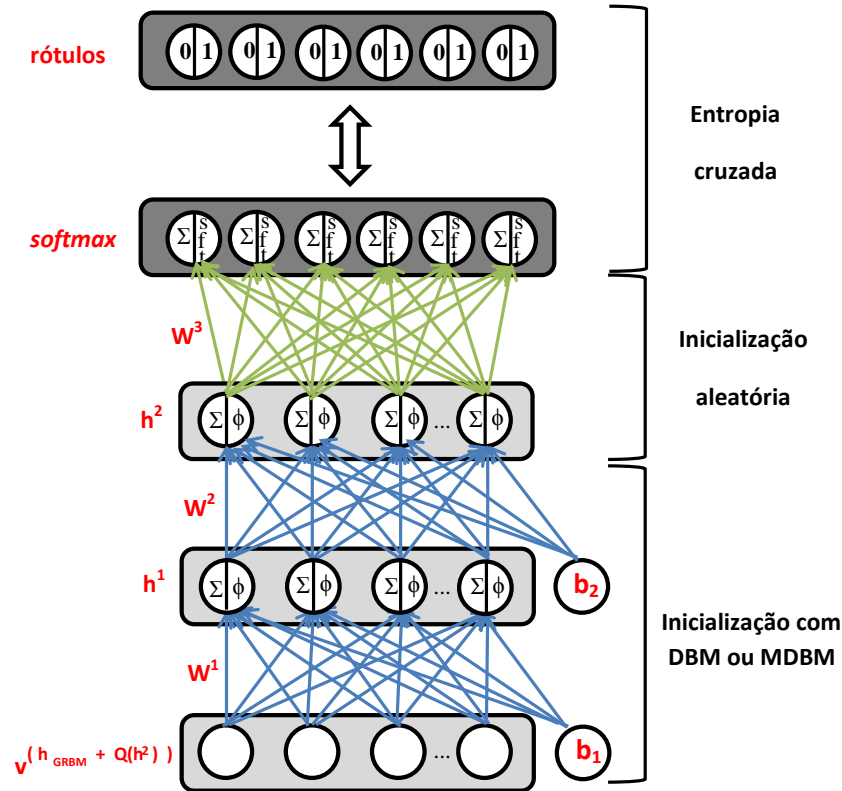
Por meio da notação $\mathbf{v}(\mathbf{h}_{\text{GRBM}} + Q(\mathbf{h}^2))$ é destacado na Figura 11 que as imagens de entrada utilizadas no treinamento supervisionado deverão estar na forma pré-processada, sendo que \mathbf{h}_{GRBM} simboliza a camada \mathbf{h} da GRBM gerada pelo pré-treinamento global e $Q(\mathbf{h}^2)$ denota os valores da camada \mathbf{h}^2 após aplicação de no máximo 50 interações de MF sobre a rede MLP recém inicializada.

Na Figura 11 também recebe destaque a função aplicada sobre a camada de saída da rede neural MLP (camada *softmax* da mesma figura), sendo esta função representada por:

$$\text{softmax}(\mathbf{h}^2) = \frac{e^{(\mathbf{h}^2 \mathbf{W} + \mathbf{b})}}{\sum_i e^{(h_i^2 W_{ij} + b_i)}}. \quad (28)$$

A Equação (28) é utilizada para normalizar o resultado do produto interno

Figura 11 – Inicialização de MLP.



Fonte: Elaborada pelo autor.

calculado entre \mathbf{h}^2 e \mathbf{W}^3 sendo que estes são comparados posteriormente com os vetores de rótulos binários por meio da Entropia Cruzada denotada por:

$$H(O, S) = - \sum_i O_i \log S(z_i), \quad (29)$$

onde $S(\mathbf{z})$ representa a distribuição de probabilidades calculada sobre $\mathbf{z} = \text{softmax}(\mathbf{h}^2)$ e O representa os rótulos das imagens de treinamento.

Após a etapa de treinamento, para utilizar o classificador, em um primeiro momento é aplicado sobre a imagem de teste (em escala de $32 \times 32 \text{ pixels}$) o processo de subdivisão de regiões de tamanho 8×8 (incluindo o processamento da vigésima sexta região); em um segundo momento as 26 regiões são concatenadas de modo a formar um único vetor unidimensional; em um terceiro momento este vetor unidimensional é submetido para a GRBM gerada no pré-treinamento global da Seção 5.2; em um quarto momento, após gerar $Q(\mathbf{h}^2)$, o vetor de entrada $\mathbf{v}^{(\mathbf{h}+Q(\mathbf{h}^2))}$ é submetido para o classificador MLP que retorna na

camada *softmax* da Figura 11 o resultado. A posição do nó de maior valor de ativação da camada *softmax* irá determinar a classe do veículo presente na imagem de teste.

6 Material e Metodologia de Avaliação

Este capítulo apresenta os recursos computacionais, a base de dados e a metodologia de avaliação adotados neste trabalho para o desenvolvimento do método proposto e sua avaliação. A Seção 6.1 mostra as bibliotecas de funções que foram utilizadas como base para construção do reconhecedor de veículos, a Seção 6.2 descreve todas as características da base de dados utilizada nos experimentos realizados para avaliar o reconhecedor e a Seção 6.3 descreve a metodologia de avaliação do método proposto.

6.1 Recursos Computacionais

Foram utilizadas como base para a codificação dos algoritmos as funções em Matlab disponibilizadas respectivamente em (SALAKHUTDINOV, 2012) e em (SALAKHUTDINOV; HINTON, 2006). Com o intuito de acelerar a execução das etapas de pré-treinamento local e global da rede neural MLP, foram utilizados recursos de GPU (*Graphics Processing Unit*) por meio de programação em CUDA (*Compute Unified Device Architecture*) para realizar as operações de multiplicação de matrizes e amostragem de valores. Também foram utilizados outros recursos de programação, tais como a biblioteca de funções OpenCV (*Open Source Computer Vision Library*) descrita no trabalho de Bradski (2000), a biblioteca BLAS (*Basic Linear Algebra Subprograms*) e *Boost*. Como recursos de hardware, foi utilizado um computador portátil com 8 GB de memória RAM, processador intel i7-3630QM e placa gráfica NVIDIA GEFORCE GT 650M com 2GB de memória.

6.2 Base de Dados BIT-Vehicle

Para avaliar o método proposto nesta dissertação de mestrado, foi utilizada a base BIT-Vehicle (*Beijing Institute of Technology*), criada por Dong et al. (2015). Esta base contém 9905 imagens de veículos em tamanhos de 1600 x 1200 e 1920 x 1080 *pixels* capturadas por duas câmeras diferentes em tempos distintos posicionadas em ângulos e lugares também diferentes. As condições de luminosidade, escala, cor de superfícies são variáveis de uma imagem para outra e devido à atrasos de captura das câmeras e variação

de tamanho de alguns veículo, em algumas imagens partes destes veículos não aparecem. A Figura 12 apresenta algumas amostras da base de dados BIT-Vehicle (DONG et al., 2015).

Figura 12 – Base de Dados BIT-Vehicle.



Fonte: Elaborada pelo autor.

Os veículos da base BIT-Vehicle são distribuídos em 6 classes, sendo elas:

- Ônibus, com 555 imagens;
- Micro-ônibus, com 878 imagens;
- Minivan, com 474 imagens;
- Sedan, com 5796 imagens;
- SUV, com 1381 imagens;
- Caminhão, com 821 imagens.

Como existem imagens contendo mais de um veículo, as coordenadas superior esquerda e inferior direita das regiões com veículos foram pré-annotadas.

6.3 Metodologia de Avaliação do Método Proposto

Tendo em vista a dificuldade para se encontrar bases públicas de imagens de veículos que possuam uma variedade razoável de classes, neste trabalho foram utilizados dois grupos de avaliação denominados *Grupo de Avaliação de Classes Não Combinadas* e *Grupo de Avaliação de Classes Combinadas*, conforme descrito nas Subseções 6.3.1 e 6.3.2, ambos os grupos obtidos da base de dados BIT-Vehicle.

O método de avaliação adotado para os testes pertencentes a cada grupo foi a acurácia média, sendo esta a mesma forma de avaliação adotada por Dong et al. (2015). O resultado de acurácia média foi produzido após a execução de cinco rodadas¹ de avaliação (treinamento e teste).

6.3.1 Grupo de Avaliação de Classes Não Combinadas

Neste grupo composto por três diferentes tipos de experimentos todas as seis classes de veículos, vide Seção 6.2, foram utilizadas nos processos de pré-treinamento, treinamento e reconhecimento da rede neural MLP de reconhecimento de veículos. Os experimentos dividiram-se em: Teste com Base Original (**TO**), Teste com Base Normalizada (**TN**) e Teste com Base Aumentada (**TA**), descritos nas Subseções 6.3.1.1, 6.3.1.2 e 6.3.1.3.

¹ A cada nova rodada de avaliação do método novos subconjuntos de treino e teste são formados utilizando-se randomização. Este método permite repetição de imagens nos subconjuntos de avaliação, diferentemente do método k-fold por exemplo.

6.3.1.1 Teste com Base Original

Neste teste, exceto a redução das imagens para o tamanho de 32×32 pixels, nenhum outro tipo de pré-processamento foi aplicado na base BIT-Vehicle.

Após a redução das imagens, 2400 imagens foram randomicamente extraídas da base e separadas em dois conjuntos de 1200 imagens (conjunto de treino e conjunto de teste), sendo cada conjunto formado por 200 imagens de cada classe de veículo. Este processo foi repetido por mais quatro vezes e ao final foi calculado o resultado da acurácia média sobre os conjuntos de teste.

6.3.1.2 Teste com Base Normalizada

Neste teste, com o objetivo de superar problemas de luminosidade encontrados em muitas imagens da base BIT-Vehicle, normalizou-se os conjuntos de treino e teste por meio da aplicação combinada de duas técnicas de equalização de histogramas baseadas na decomposição dos espaços de cores HSV (*Hue Saturation Value*) e Lab das imagens, vide (WARE, 2012) e (GONZALEZ; WOODS, 2006). A primeira técnica consistiu na conversão das imagens do espaço de cores RGB para o espaço de cores HSV e, posteriormente, na aplicação da equalização do histograma gerado para a componente V (espaço de cores HSV), tendo em vista que esta componente corresponde à intensidade luminosa da imagem no espaço de cores HSV (esta técnica mostrou-se muito eficaz para a melhoria de contraste das regiões escuras das imagens RGB). Como a técnica de equalização de imagens HSV não foi capaz de melhorar a qualidade de todas as imagens da base BIT-Vehicle (em especial nos casos de muita luminosidade) foi aplicado em contrapartida o método CLAHE (*Contrast Limited Adaptive Histogram Equalization*) criado por Zuiderveld (1994).

Para determinar qual técnica deveria ser aplicada utilizou-se a análise da assimetria s do histograma da componente V do espaço de cores HSV, calculada pela Equação (30):

$$s(V) = \frac{E_p(V - \mu)^3}{\sigma^3}, \quad (30)$$

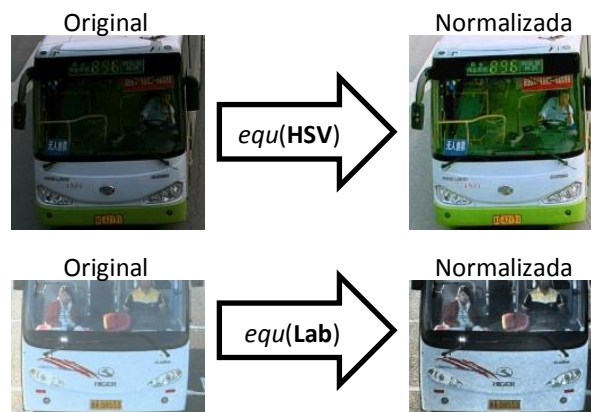
onde $E_p(\cdot)$ indica a esperança matemática, V o conjunto de *pixels*, μ e σ respectivamente o valor de média e o desvio padrão calculados sobre o conjunto V .

A normalização por CLAHE foi aplicada em detrimento da normalização por HSV sempre que $s(V)$ era maior do que 1,0. A Figura 13 mostra os resultados da aplicação de ambas as técnicas de equalização em circunstâncias onde os valores de assimetrias

negativas e positivas do histograma da componente V foram acentuadas, observando que *equ* denota o tipo de equalização aplicada.

Assim como no primeiro teste, as imagens foram reduzidas para o tamanho de 32×32 pixels e agrupadas nos conjuntos de treino e teste, como descrito na Subseção 6.3.1.1.

Figura 13 – Normalização de imagens de veículos.



Fonte: Elaborada pelo autor.

6.3.1.3 Teste com Base Aumentada

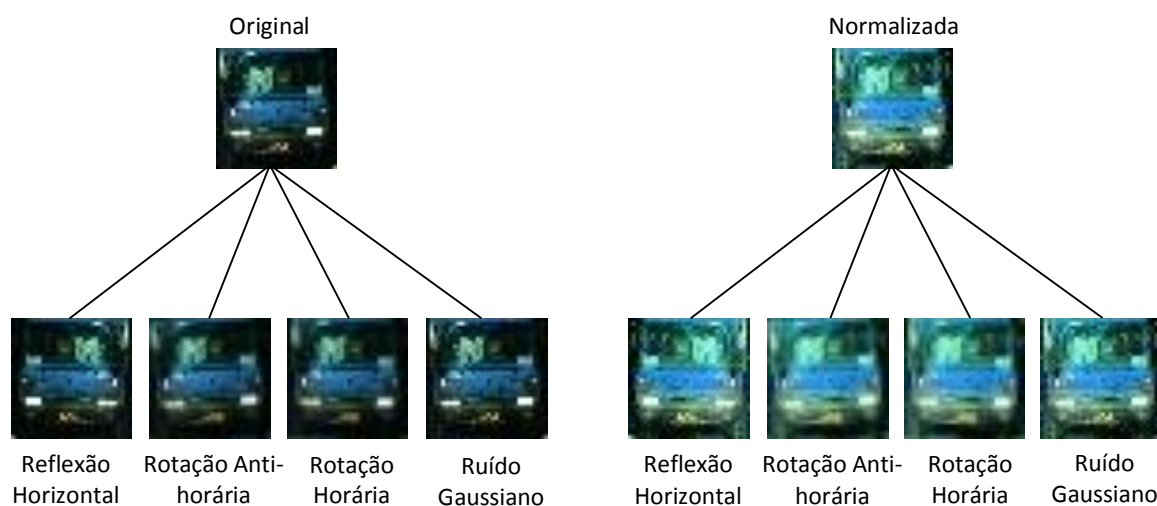
Por ser uma técnica relativamente simples e eficiente, o aumento artificial do conjunto de imagens de treinamento foi uma estratégia utilizada neste trabalho para solucionar principalmente problemas de *overfitting* (casos em que a rede neural aprende muito bem o modelo de distribuição da base de treinamento, mas encontra deficiências de generalização para dados externos ao treinamento), que são muito comuns em métodos de aprendizado de máquinas baseados em redes neurais, vide (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Neste trabalho foi adotada uma estratégia de aumento artificial das imagens de treinamento que consistiu primeiramente em determinar a classe com maior número de imagens disponíveis, neste caso a classe Sedan com um total de 5796 imagens. De um conjunto de 2498² imagens, 200 imagens foram randomicamente selecionadas para

² Para obter o conjunto de 2498 imagens, primeiro dividiu-se o valor de 5796 pela metade e posteriormente

a construção do conjunto de teste e as 2298 imagens restantes foram utilizadas como referência³. Sendo assim, para aumentar a quantidade de imagens da classe Ônibus, por exemplo, foi necessário aplicar seis diferentes⁴ tipos de transformações (considerando-se neste caso que a quantidade final de imagens de Ônibus precisava ser no mínimo igual a quantidade de Sedans). Este cálculo foi feito para aproximar a quantidade de 355 imagens de Ônibus (excluídas as 200 imagens de teste) da quantidade de 2298 imagens de Sedans. Além da transformação já mencionada na Seção 6.3.1.2, foram utilizadas quatro outras transformações: reflexão horizontal, rotação de 2° nos sentidos horário e anti-horário e adição de ruído branco gaussiano $\mathcal{N}(0, 10^{-6})$.

Figura 14 – Aumento artificial de base de dados.



Fonte: Elaborada pelo autor.

A Figura 14 mostra um caso de aumento artificial da base de dados onde todas as transformações tiveram que ser aplicadas, neste caso para a classe das Minivans. Assim como no Teste com Base Original e no Teste com Base Normalizada a mesma estratégia de geração do conjunto de imagens de teste foi utilizada, porém a quantidade de imagens do conjunto de treinamento foi, neste teste, 15081.

fora subtraído do resultado o valor de 400. Mesmo sem maiores evidências, este método mostrou-se bastante eficaz como solução automática para aumento artificial de bases.

³ É necessário haver uma classe referência para poder determinar quantas transformações devem ser aplicadas sobre as imagens das outras classes de veículos.

⁴ Apesar das transformações aplicadas serem diferentes é necessário que as imagens transformadas sejam parecidas entre si para não gerar problemas de *underfitting* ao treinar a rede neural MLP.

6.3.2 Grupo de Avaliação de Classes Combinadas

Neste grupo de avaliação as seis classes de veículos da base BIT-Vehicle foram combinadas em três classes, de acordo com o porte dos veículos. Seguindo esta lógica foram agrupados Ônibus e Micro-ônibus na classe Ônibus, Sedans e SUVs na classe Carros e, por fim, Caminhões e Minivans na classe Caminhões. A classe Minivan foi combinada com a classe Caminhão pois as imagens de Minivans da base BIT-Vehicle possuem uma pequena carroceria para transporte de cargas.

Os mesmos três tipos de experimentos do Grupo de Avaliação de Classes Não Combinadas, descritos na Subseção 6.3.1, foram aqui utilizados, porém devido à combinação das classes a metodologia de construção dos conjuntos de treino e teste foi modificada, sendo que a quantidade de imagens de treino e teste coletadas de cada classe passou a ser 400 (totalizando 1200 imagens nos conjuntos de treino e teste do **TO** e **TN** e conjunto de teste do **TA**).

O valor de referência para aplicação da estratégia de aumento de base passou a ser 3188, correspondendo à quantidade de imagens na classe Carros. O conjunto de treinamento passou a ter 9867 imagens após terem sido aplicadas no máximo três das técnicas de aumento de base (vide Subseção 6.3.1.3) nas classes Caminhões e Ônibus, sendo elas a normalização (explicada na Seção 6.3.1.2), a reflexão horizontal da imagem original e a reflexão horizontal da imagem normalizada.

7 Resultados e Discussão

Neste capítulo são apresentados os resultados obtidos pelos classificadores baseados nas abordagens 2D-DBM (*Bilinear Deep Boltzmann Machine*) e 2D-MDBM (*Bilinear Multinomial Deep Boltzmann Machine*), ambas propostas nesta dissertação de mestrado, para o reconhecimento de veículos em imagens coloridas. Tais resultados são comparados com os resultados obtidos por classificadores baseados em DBM e MDBM e também com classificadores baseados em redes neurais convolucionais.

Tendo em vista que durante a fase de pré-treinamento local (vide Seção 5.1) a quantidade máxima de *pixels* por canal RGB das imagens de tamanho 8×8 é 64, adotou-se que a quantidade máxima de neurônios utilizados na camada escondida da GRBM (*Gaussian restricted Boltzmann Machine*) e nas duas camadas escondidas de cada estrutura profunda (DBM ou MDBM) de pré-treinamento local também seria 64. O critério de escolha do número de camadas foi baseado na demonstração feita por Salakhutdinov e Hinton (2012) de que para DBMs com até duas camadas é garantido que o limite variacional é aumentado após aplicação dos treinamentos em gride e por MF (*Mean Field*) da estrutura de rede profunda.

Sobre os classificadores 2D-DBM e 2D-MDBM foram aplicadas 80 épocas de pré-treinamento local e global à uma taxa de aprendizado de 0,02 para os *biases* \mathbf{c} e \mathbf{b} e de 0,01 para o parâmetro \mathbf{W} na etapa de treinamento de GRBMs locais. Para o treinamento das estruturas profundas locais (DBMs e MDBMs) estes valores foram aumentados para 0,2 e 0,1. Nos pré-treinamentos globais em gride as taxas de aprendizado foram reduzidas para 0,001 e 0,002 durante o pré-treinamento da GRBM e aumentados para 0,01 e 0,02 durante o pré-treinamento global em gride das camadas escondidas. Também foi aplicado um *momentum* de 0,5 até a quinta época e de 0,9 da sexta até a octogésima época durante os pré-treinamentos local e global em gride. A aproximação estocástica por MF (vide Seção 2.2.4) foi executada por 50 épocas à uma taxa de aprendizado de 0,0005 e *momentums* de 0,1 da primeira até a quinta época e de 0,5 da sexta até a quinquagésima época utilizando uma quantidade máxima de 30 interações para atingir o equilíbrio térmico. É importante mencionar que o treinamento utilizando MF só foi executado durante a fase de pré-treinamento global. A mesma configuração de pré-treinamento global da 2D-DBM

e 2D-MDBM foi utilizada para pré-treinar os classificadores DBM e MDBM.

Com relação ao treinamento dos classificadores utilizando gradiente conjugado, as únicas modificações¹ feitas no código original de foram a mudança da quantidade máxima de interação do método de buscas em reta e a quantidade de épocas de execução. Foi utilizado em um primeiro momento um valor máximo de dez buscas para os parâmetros da rede neural MLP inicializados aleatoriamente, vide Figura 11 e em um segundo momento, após cinco interações, este valor foi modificado para 5. A quantidade utilizada de épocas de execução do método foi de 50 para os classificadores (DBM, MDBM, 2D-DBM e 2D-MDBM) exceto nos **TAs** de ambos os grupos de avaliação de resultados, onde a quantidade de épocas de execução dos métodos 2D-DBM e 2D-MDBM foi aumentada² para 100.

Nas etapas de pré-treinamento foram utilizados lotes de tamanho 100 e nas etapas de treinamento utilizando gradiente conjugado foram utilizados lotes de tamanho 400, para os **TOs** e **TNs** de ambos os grupos de avaliação; lotes de tamanho 490 para o **TA** do Grupo de Avaliação de Classes Não Combinadas; e lotes de tamanho 400 para o **TA** do Grupo de Avaliação de Classes Combinadas.

7.1 Resultados do Grupo de Avaliação de Classes Não Combinadas

A Tabela 1 apresenta os resultados das acurácias médias obtidas por meio do Teste com Base Original (**TO**), do Teste com Base Normalizada (**TN**) e do Teste com Base Aumentada (**TA**) pelos classificadores MLP pré-treinados utilizando apenas redes profundas (DBM e MDBM) e pelos classificadores gerados por ambas as abordagens incluídas na proposta deste trabalho (2D-DBM e 2D-MDBM). A Tabela 1 mostra que os resultados obtidos pelos classificadores 2D-DBM e 2D-MDBM foram muito próximos aos resultados obtidos pelos seus métodos correspondentes, DBM e MDBM. A maior diferença foi de 2,83% na comparação do 2D-DBM com o DBM no **TO** e a menor diferença foi de 1,58% obtida também por meio da comparação entre o 2D-DBM e o DBM, porém no **TA**.

Por meio da Tabela 1 é possível notar também que os resultados obtidos pelos classificadores foram melhores quando aplicada a normalização e o aumento de base.

¹ O código de Salakhutdinov (2012) fora inicialmente desenvolvido para solucionar o problema de reconhecimento de dígitos manuscritos em imagens em tons de cinza (muito próximas de imagens binárias). Por necessidade de adaptação do código original ao problema de reconhecimento de imagens de veículos é que foram feitas algumas modificações.

² Este aumento foi necessário pois observou-se durante os **TAs** que o gradiente conjugado estava convergindo por volta da octogésima época.

Também foi possível notar que o **TA** provocou um aumento de 4,31% na acurácia média do método 2D-MDBM em comparação com o **TO**, sendo este o maior aumento observado entre os quatro métodos.

Tabela 1 – Acurácias médias em porcentagem do Grupo de Avaliação de Classes Não Combinadas.

MÉTODOS	TO	TN	TA
DBM	80,03 ± 0,02	80,62 ± 0,03	83,00 ± 0,01
MDBM	80,26 ± 0,02	80,57 ± 0,02	83,75 ± 0,01
2D-DBM	77,20 ± 2,80	78,95 ± 2,79	81,42 ± 1,35
2D-MDBM	77,52 ± 3,39	78,97 ± 3,04	81,83 ± 1,47

Fonte: Elaborada pelo autor

A Tabela 2 mostra que os tempos médios de execução do treinamento dos métodos 2D-DBM e 2D-MDBM foram muito menores em comparação com os métodos DBM e MDBM, respectivamente. No melhor caso, obtido na comparação entre os métodos DBM e 2D-DBM no **TN**, o tempo de treinamento foi 5,5 vezes menor. No pior caso, obtido na comparação entre os métodos DBM e 2D-DBM no **TA**, o tempo foi 2 vezes menor.

Tabela 2 – Tempos médios de treinamento em minutos do Grupo de Avaliação de Classes Não Combinadas.

MÉTODOS	TO	TN	TA
DBM	44	93	288
MDBM	44	93	582
2D-DBM	15	17	143
2D-MDBM	15	20	210

Fonte: Elaborada pelo autor

Na Tabela 2 é possível verificar também que os tempos médios de execução dos métodos DBM e MDBM foram 2,11 vezes maiores no **TN** em comparação com o **TO**. Durante o treinamento destes métodos foi observado que este comportamento estava associado com a quantidade máxima de iterações exigidas pelo método de gradiente conjugado, tendo em vista que no **TO** foram necessárias, na maioria das vezes, menos de cinco buscas em reta por lote de imagens, para o método convergir. Este também foi

o principal motivo para o aumento dos tempos de execução do **TA**, além da influência exercida pelo aumento da quantidade de imagens de treinamento. Deve-se levar em conta também que o método de gradiente conjugado disponibilizado em (SALAKHUTDINOV; HINTON, 2006) não pôde ser executado de forma paralelizada em GPU.

7.2 Resultados do Grupo de Avaliação de Classes Combinadas

A Tabela 3, de forma análoga à Tabela 1, mostra as acurácias médias obtidas no **TO** **TN** e **TA** para a base combinada em Carros, Ônibus e Caminhões e a Tabela 4, de forma análoga à Tabela 2, mostra os tempos médios despendidos pelo pré-treinamento e treinamento dos métodos DBM, MDBM, 2D-DBM e 2D-MDBM.

Tabela 3 – Acurácias médias em porcentagem do Grupo de Avaliação de Classes Combinadas.

MÉTODOS	TO	TN	TA
DBM	88,45 ± 0,01	89,50 ± 0,01	93,13 ± 0,01
MDBM	88,18 ± 0,01	89,18 ± 0,01	93,20 ± 0,01
2D-DBM	85,53 ± 1,16	87,15 ± 1,43	91,13 ± 0,95
2D-MDBM	86,28 ± 0,26	88,12 ± 1,07	91,10 ± 0,60

Fonte: Elaborada pelo autor

As acurácias médias aumentaram significativamente ao serem combinadas as classes. Comparando-se os **TOs** da Tabela 3 com os da Tabela 1 verifica-se que no pior caso, métodos MDBMs, o aumento da acurácia média foi de 7,92%, sendo que no melhor caso, métodos 2D-MDBMs, este aumento foi de 8,76%. Comparando-se os **TNs** no pior caso, métodos 2D-DBMs, o aumento foi de 6,2% e no melhor caso, métodos DBMs, o aumento foi de 8,88%. Comparando-se os **TAs** no pior caso, métodos 2D-MDBMs, o aumento foi de 9,27% e no melhor caso, métodos DBMs, o aumento foi de 10,13%.

A Tabela 3, assim como a Tabela 1, mostra que os resultados obtidos pelos classificadores 2D-DBM e 2D-MDBM foram muito próximos aos resultados obtidos pelos classificadores correspondentes DBM e MDBM. A maior diferença foi de 2,92% na comparação do 2D-DBM com o DBM no **TO** e a menor diferença foi de 1,06% obtida por meio da comparação entre o 2D-MDBM e o MDBM no **TN**.

Por meio da Tabela 3 é possível notar também que os resultados obtidos por todos os métodos mais uma vez foram melhores do que o **TO** tanto no **TN** quanto no **TA** e que o aumento mais significativo foi de 5,6% para o método 2D-DBM no **TA**.

A Tabela 4, assim como a Tabela 2, mostra que os tempos médios de execução do treinamento dos métodos 2D-DBM e 2D-MDBM foram mais uma vez muito menores em comparação com os métodos DBM e MDBM respectivamente. No melhor caso, obtido na comparação entre os métodos DBM e 2D-DBM no **TN**, o tempo de treinamento foi 5 vezes menor. No pior caso, obtido na comparação entre os métodos DBM e 2D-DBM no **TA**, o tempo foi 2,92 vezes menor.

Tabela 4 – Tempos médios de treinamento em minutos do Grupo de Avaliação de Classes Combinadas.

MÉTODOS	TO	TN	TA
DBM	30	95	193
MDBM	45	98	203
2D-DBM	10	19	66
2D-MDBM	10	20	69

Fonte: Elaborada pelo autor

7.3 Resultados Obtidos com CNN

Foram comparados também os resultados de acurácia média obtidos pelos classificadores 2D-DBM e 2D-MDBM nos testes **TO** e **TN**, do Grupo de Avaliação de Classes Não Combinadas da Subseção 6.3.1, com a acurácia média obtida pelo método de classificação criado por Dong et al. (2015). Partindo do princípio de que o método de Dong et al. (2015) baseia-se em CNNs, comparou-se o mesmo com o método de classificação de objetos em imagens coloridas desenvolvido por Krizhevsky (2014). O método CNN de Krizhevsky (2014) foi executado utilizando os parâmetros originais.

Analisando os resultados dos métodos CNNs foi possível observar que o método de Krizhevsky (2014) obteve seu melhor resultado no **TA**, sendo este valor igual à 85,65% com desvio padrão de 0,01%, e seu pior resultado no **TN**, sendo este igual à 69,33% com desvio padrão de 0,02%. Quando comparado com o valor de 88% obtido por Dong et al. (2015) pode-se notar que a CNN de Krizhevsky (2014) mesmo sem utilizar qualquer

tipo de pré-treinamento conseguiu obter um resultado apenas 2,35% menor que a CNN pré-treinada de Dong et al. (2015), validando ainda mais a técnica de aumento de base adotada nesta dissertação de mestrado.

8 Conclusão e Sugestões para Trabalhos Futuros

Neste trabalho foi proposta uma nova forma eficiente de pré-treinamento de MLPs utilizando projeção bilinear e duas diferentes abordagens de aprendizado em profundidade, DBM (*Deep Boltzmann Machine*) e MDBM (*Multinomial Deep Boltzmann Machine*). A abordagem de pré-treinamento com DBM produziu o classificador 2D-DBM (*Bilinear Deep Boltzmann Machine*) e a abordagem MDBM (*Bilinear Multinomial Deep Boltzmann Machine*) produziu o classificador 2D-MDBM, que após serem treinados utilizando gradiente conjugado, provaram ser eficientes na tarefa de reconhecimento de veículos em imagens coloridas.

Os dois classificadores mostraram-se eficientes na tarefa de reconhecimento de seis e três diferentes classes de veículos, considerando neste caso respectivamente a análise feita na base BIT-Vehicle original e na sua versão de classes combinadas.

Em termos de acurácias médias as diferenças entre os classificadores 2D-DBM e 2D-MDBM e os classificadores DBM e MDBM mantiveram-se abaixo de 3%. Em termos de tempo médio de execução de pré-treinamento e treinamento os métodos 2D-DBM e 2D-MDBM apresentaram um ganho bastante expressivo, onde no melhor caso o tempo médio de treinamento do classificador 2D-DBM foi 5,5 vezes menor em comparação com a DBM.

A primeira grande vantagem dos métodos 2D-DBM e 2D-MDBM é a diminuição no tempo de treinamento sem perda significativa de acurácia. A segunda vantagem é uma consequência da primeira e diz respeito à possibilidade de utilização do 2D-DBM e do 2D-MDBM em tarefas de otimização de parâmetros de pré-treinamento e treinamento da MLP. A terceira vantagem desta proposta é que ambos os classificadores 2D-DBM e 2D-MDBM podem facilmente ser utilizados em outros tipos de problemas discriminativos envolvendo imagens do mundo real.

8.1 Contribuições deste Trabalho

Este trabalho contribuiu com o desenvolvimento de dois classificadores eficientes (2D-DBM e 2D-MDBM) que podem ser aplicados em sistemas ITS que necessitem reconhecer veículos, ou em outras classes de problemas que envolvam imagens do mundo real. Outras contribuições foram:

- O desenvolvimento de um método eficaz de projeção bilinear de regiões de imagens coloridas, como mostra a Seção 5.1;
- O desenvolvimento de um método de normalização de imagens combinando equalização de imagens coloridas em HSV com equalização de imagens coloridas em Lab;
- O desenvolvimento de um método para aumentar bases de treinamento reduzindo *overfitting*.

8.2 Propostas para Trabalhos Futuros

Os métodos 2D-DBM e 2D-MDBM, desenvolvidos nesta dissertação de mestrado, poderão ser aplicados em problemas de generalização envolvendo restauração de imagens e até mesmo compactação de informação, como no caso dos auto-encodificadores criados por Hinton e Salakhutdinov (2006). Também poderia se analisar a capacidade de aplicação da técnica de pré-treinamento em CNNs. Outra sugestão, neste caso relacionada à engenharia de tráfego, seria acoplar o método de reconhecimento já treinado à um ITS de contagem classificatória de veículos.

8.3 Trabalhos Publicados e Submetidos

Trabalhos aceitos para publicação:

1. PIRES, R. G.; SANTOS, D. F. S.; SOUZA, G. B.; LEVADA, A. L. M.; PAPA, J. P. A Deep Boltzmann Machine-Based Approach For Robust Image Denoising. In: *Iberoamerican Congress on Pattern Recognition - CIARP*, 2017, Valparaíso - Chile.

2. PIRES, R. G.; LEVADA, A. L. M.; SOUZA, G. B.; PEREIRA, L. A. M.; SANTOS, D. F. S.; PAPA, J. P. A Robust Restricted Boltzmann Machine for Binary Image Denoising. In: *Conference on Graphics, Patterns and Images - SIBGRAPI*, 2017, Niterói - Rio de Janeiro.
3. SANTOS, D. F. S.; SOUZA, G. B.; MARANA, A. N. A 2D Deep Boltzmann Machine for Robust and Fast Vehicle Classification. In: *Conference on Graphics, Patterns and Images - SIBGRAPI*, 2017, Niterói - Rio de Janeiro.
4. SOUZA, G. B.; SANTOS, D. F. S.; PIRES, R. G.; MARANA, A. N.; PAPA, J.P. A Novel LBP-Based Convolution Neural Network for Robust Facial Spoofing Detection. In: *IEEE International Symposium on Circuits & Systems - ISCAS*, 2017, Baltimore, USA.
5. SOUZA, G. B.; SANTOS, D. F. S.; PIRES, R. G.; MARANA, A. N.; PAPA, J.P. Deep Boltzmann Machines for Robust Fingerprint Spoofing Attack Detection. In: *International Joint Conference on Neural Networks - IJCNN*, 2017, Anchorage, Alaska, USA.
6. SOUZA, G. B.; SANTOS, D. F. S.; PIRES, R. G.; MARANA, A. N.; PAPA, J. P. Efficient Transfer Learning for Robust Face Spoofing Detection. In: *Iberoamerican Congress on Pattern Recognition - CIARP*, 2017, Valparaiso - Chile.

Trabalhos submetidos para publicação:

1. SOUZA, G. B.; SANTOS, D. F. S.; PIRES, R. G.; MARANA, A. N.; PAPA, J. P. A Restricted Boltzmann Machine-Based Approach for Robust Dimensionality Reduction. In: *XIII Workshop de Visão Computacional - WVC 2017*, 2017, Natal - RN.

REFERÊNCIAS

- ALMAGAMBETOV, A.; S., V.; CASARES, M. Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras. *IEEE Transactions on Industrial Electronics*, IEEE, v. 62, n. 6, p. 3732–3741, 2015.
- BAEK, Y.; KIM, W. Forward vehicle detection using cluster-based adaboost. *Optical Engineering*, SPIE, v. 53, n. 10, p. 102103–102117, 2014.
- BENGIO, Y. Learning deep architectures for ai. *Foundations and trends[®] in Machine Learning*, Now Publishers Inc., v. 2, n. 1, p. 1–127, 2009.
- BIRRELL, S. A.; FOWKES, M.; JENNINGS, P. A. Effect of using an in-vehicle smart driving aid on real-world driver performance. *IEEE Transactions on Intelligent Transportation Systems*, IEEE Journals & Magazines, v. 15, n. 4, p. 1801–1810, 2014.
- BRADSKI, G. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, Miller Freeman Inc., v. 25, n. 11, p. 120–123, 2000.
- CARREIRA-PERPIÑÁN, M. Á.; HINTON, G. E. On contrastive divergence learning. In: AISTATS 2005 PROCEEDINGS OF THE TENTH INTERNATIONAL WORKSHOP ON ARTIFICIAL INTELLIGENCE AND STATISTICS, 2005, Hotel Savannah, Barbados. *Anais...* Hotel Savannah, Barbados: Society for Artificial Intelligence and Statistics, 2005. p. 33–40.
- DAYAN, P.; ABBOTT, L. F. *Theoretical Neuroscience: Computational and mathematical modeling of neural systems*. 6. ed. One Rogers Street, Cambridge MA 02142-1209, US: The MIT Press, 2005.
- DIESTEL, R. *Graph Theory*: Electronic edition 2005. 3. ed. New York: Springer-Verlag Heidelberg, 2005.
- DONG, Z.; WU, Y.; PEI, M.; Y., J. Vehicle type classification using a semisupervised convolution neural network. *IEEE Transactions on Intelligent Transportation Systems*, IEEE Journals & Magazines, v. 16, n. 4, p. 2247–2256, 2015.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 3. ed. Upper Saddle River, NJ, EUA: Prentice-Hall, Inc., 2006.
- HE, X.; NIYOGI, P. Locality preserving projections. In: THRUN, S.; SAUL, L. K.; SCHÖLKOPF, P. B. (Ed.). *Advances in Neural Information Processing Systems 16*. EUA, Cambridge: MIT Press, 2004. p. 153–160.
- HINTON, G.; KRIZHEVSKY, A.; NAIR, V. *CIFAR-10 and CIFAR-100 datasets*. 2009. Disponível em: <<https://www.cs.toronto.edu/~kriz/cifar.html>>. Acesso em: 20 jul. 2016.
- HINTON, G. E.; OSINDERO, S.; TEH, Y. A fast learning algorithm for deep belief nets. *Neural computation*, MIT Press, v. 18, n. 7, p. 1527–1554, 2006.

- HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science*, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.
- HU, A.; LI, H.; ZHANG, F.; ZHANG, W. Deep Boltzmann machines based vehicle recognition. In: THE 26TH CHINESE CONTROL AND DECISION CONFERENCE (2014 CCDC), 2014, Changsha, China. *Anais...* Changsha, China: IEEE, 2014. p. 3033–3038.
- KRIZHEVSKY, A. *Learning Multiple Layers of Features from Tiny Images*. 1. ed. 27 King's College Cir, Toronto, on M5S, Canada: Universidade de Toronto, 2009.
- KRIZHEVSKY, A. *cuda-convnet*. 2014. Disponível em: <<https://code.google.com/p/cuda-convnet/>>.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGESS, C. J. C.; BOTTOU, L.; WEINBERGER, K. Q. (Ed.). *Advances in Neural Information Processing Systems 25*. 1. ed. Nova York, EUA: Curran Associates, Inc., 2012. p. 1097–1105.
- LECUN, Y.; BOTTOU, L.; ORR, G. B.; MÜLLER, K.-R. Efficient backprop. In: MONTAVON, G.; ORR, G. B.; MÜLLER, K.-R. (Ed.). *Neural Networks: Tricks of the Trade*. 2. ed. Heidelberg: Springer, 2012. p. 9–48.
- LIANG, Z.; LI, Y.; SHI, P. A note on two-dimensional linear discriminant analysis. *Pattern Recogn. Lett.*, Elsevier Science Inc., Nova York, EUA, v. 29, n. 16, p. 2122–2128, dec 2008.
- LU, G.; KONG, L.; WANG, Y.; TIAN, D. Vehicle trajectory extraction by simple two-dimensional model matching at low camera angles in intersection. *IET Intelligent Transportation Systems*, IET, v. 8, n. 7, p. 631–638, 2014.
- LUCE, R. D.; PERRY, A. D. A method of matrix analysis of group structure. *Psychometrika*, Springer, v. 14, n. 2, p. 95–116, 1949.
- LV, Y.; DUAN, Y.; KANG, W.; LI, Z.; WANG, F. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 16, n. 2, p. 865–873, 2015.
- MOORE, D. S.; MCCABE, G. P.; CRAIG, B. A. *Introduction to the Practice of Statistics*. 6. ed. 41 Madison Avenue, New York, NY 10010: W. H. Freeman and Company, 2009.
- NAIR, V.; HINTON, G. E. Implicit mixtures of restricted Boltzmann machines. In: KOLLER, D.; SCHUURMANS, D.; BENGIO, Y.; BOTTOU, L. (Ed.). *Advances in Neural Information Processing Systems 21*. 1. ed. Nova York, EUA: Curran Associates, Inc., 2009. p. 1145–1152.
- NAVON, I. M.; LEGLER, D. M. Conjugate-gradient methods for large-scale minimization in meteorology. *Monthly Weather Review*, American Meteorological Society, v. 115, n. 8, p. 1479–1502, 1987.
- PETERSON, C.; ANDERSON, J. R. A mean field theory learning algorithm for neural networks. *Complex Systems*, Complex Systems Publications, Inc, v. 1, n. 5, p. 995–1019, 1987.

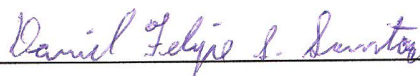
- RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, C. P. R. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. 1. ed. Cambridge, MA, EUA: MIT Press, 1986.
- SALAKHUTDINOV, R. *Learning Deep Boltzmann Machines*. 2012. Disponível em: <<http://www.cs.toronto.edu/~rsalakhu/DBM.html>>. Acesso em: 13 ago. 2016.
- SALAKHUTDINOV, R.; HINTON, G. *Training a deep autoencoder or a classifier on MNIST digits*. 2006. Disponível em: <<http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html>>. Acesso em: 13 ago. 2016.
- SALAKHUTDINOV, R.; HINTON, G. E. An efficient learning procedure for deep Boltzmann machines. *Neural Computation*, MIT Press, v. 24, n. 8, p. 1967–2006, 2012.
- SALAKHUTDINOV, R.; LAROCHELLE, H. Efficient learning of deep Boltzmann machines. In: PROCEEDINGS OF THE THIRTEENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND STATISTICS, 2010, Resort Chia Laguna, Sardinia, Italia. *Anais...* Resort Chia Laguna, Sardinia, Italia: PMLR, 2014. p. 693–700.
- SALAKHUTDINOV, R.; TENENBAUM, J. B.; TORRALBA, A. Learning with hierarchical-deep models. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 35, n. 8, p. 1958–1971, 2013.
- SALVI, G. An automated nighttime vehicle counting and detection system for traffic surveillance. In: PROCEEDINGS OF THE 2014 INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND COMPUTATIONAL INTELLIGENCE, 2014, Las Vegas, Nevada, EUA. *Anais...* Las Vegas, Nevada, EUA: IEEE, 2014. p. 131–136.
- SONG, J.; SONG, H.; WANG, W. An accurate vehicle counting approach based on block background modeling and updating. In: 2014 7TH INTERNATIONAL CONGRESS ON IMAGE AND SIGNAL PROCESSING (CISP), 2014, Delian, China. *Anais...* Delian, China: IEEE, 2014. p. 16–21.
- SUN, N.; HAN, G.; DU, K.; LIU, X.; LI, X. Person/vehicle classification based on deep belief networks. In: 2014 10TH INTERNATIONAL CONFERENCE ON NATURAL COMPUTATION (ICNC), 2014, Universidade Xiamen, China. *Anais...* Universidade Xiamen, China: IEEE, 2014. p. 113–117.
- TEOH, S. S.; BRÄUNL, T. Symmetry-based monocular vehicle detection system. *Machine Vision and Applications*, Springer-Verlag, v. 23, n. 5, p. 831–842, 2012.
- TIELEMAN, T. Training restricted Boltzmann machines using approximations to the likelihood gradient. In: PROCEEDINGS OF THE 25TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 2008, Helsinki, Finlândia. *Anais...* Helsinki, Finlândia: IEEE, 2008. p. 113–117.
- WAINWRIGHT, M. J.; JORDAN, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, Now Publishers Inc, v. 1, n. 1-2, p. 1–305, 2008.

- WANG, H.; CAI, Y.; CHEN, L. A vehicle detection algorithm based on deep belief network. *The Scientific World Journal*, Hindawi Publishing Corporation, v. 2014, n. 647380, p. 1–7, 2014.
- WARE, C. *Information Visualization: Perception for Design*. 3. ed. San Francisco, CA, EUA: Morgan Kaufmann Publishers Inc., 2012.
- WELLING, M.; HINTON, G. E. A new learning algorithm for mean field Boltzmann machines. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, 2002, Madrid, Espanha. *Anais...* Madrid, Espanha: Springer-Verlag, 2002. p. 351–357.
- YAN, S.; XU, D.; ZHANG, B.; ZHANG, H. Graph embedding a general framework for dimensionality reduction. In: 2005 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR05), 17., 2005, San Diego, CA, EUA. *Anais...* San Diego, CA, EUA: IEEE, 2005. p. 830–837.
- YANG, J.; ZHANG, D.; YONG, X.; YANG, J.-y. Two-dimensional discriminant transform for face recognition. *Pattern recognition*, Elsevier, v. 38, n. 7, p. 1125–1129, 2005.
- YE, J.; JANARDAN, R.; LI, Q. Two-dimensional linear discriminant analysis. In: SAUL, L. K.; WEISS, Y.; BOTTOU, L. (Ed.). *Advances in Neural Information Processing Systems 17*. 1. ed. EUA, Cambridge: MIT Press, 2005. p. 1569–1576.
- ZHONG, S.; YAN, L.; YANG, L. Bilinear deep learning for image classification. In: PROCEEDINGS OF THE 19TH ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA, 2011, Scottsdale, Arizona, EUA. *Anais...* Scottsdale, Arizona, EUA: ACM, 2011. p. 343–352.
- ZUIDERVELD, K. Contrast limited adaptive histogram equalization. In: HECKBERT, P. S. (Ed.). *Graphics Gems IV*. 1. ed. San Diego, CA, EUA: Academic Press Professional, Inc., 1994. p. 474–485.

TERMO DE REPRODUÇÃO XEROGRÁFICA

Autorizo a reprodução xerográfica do presente Trabalho de Conclusão, na íntegra ou em partes, para fins de pesquisa.

São José do Rio Preto, 29 / 08 / 2017



Assinatura do autor