Detecção de placas de trânsito utilizando a rede neural convolucional profunda *AlexNet*

Lucas Vital Moreira
lucasvitalmoreira@gmail.com
Universidade Federal do Espírito Santo
Elivelto Ebermam
eliveltoebermam@gmail.com
Universidade Federal do Espírito Santo

***Resumo* — Este trabalho utiliza *convolutional neural network* (CNN) profunda *AlexNet* para a detecção e localização de placas de trânsito em imagens obtidas da base de dados do desafio *The Germam Traffic Sign Detection Benchmark* (GTSDB). A CNN profunda *AlexNet* possui 60 milhões de parâmetros e 650.000 neurônios, consiste de cinco camadas convolucionais, e três camadas totalmente conectadas com um softmax final de 1000 caminho. O trabalho realiza testes utilizando 10 redes pré-treinadas e outras 10 redes sem pré-treino, e finalmente comparando os resultados.**

Index Terms—AlexNet; CNN; Detecção de Placas de Trânsitoç GTSDB; Redes Neurais Convolucionais Profundas.

I. INTRODUÇÃO

A utilização do reconhecimento visual com visão computacional evolui rapidamente devido ao desenvolvimento e aperfeiçoamento constante das redes neurais profundas. As redes neurais convolucionais profundas fazem parte deste domínio e vem apresentando resultados altamente satisfatórios por possuir grandes conjuntos de dados pré-treinados para realizar tarefas supervisionadas de classificação de objetos, podendo consequentemente serem adaptadas para detecção de objetos ou análise de cenas [1]. O principal objetivo desse trabalho está na avaliação desta capacidade das CNNs, tendo como tarefa base a detecção de placas de trânsito. Essa necessidade é justificada pelo avanço tecnológico no que diz respeito à mobilidade humana por meio de automóveis. Os processos relacionados ao ato de se locomover por vias públicas em automóveis autônomos devem ser necessariamente, automáticos, e a tarefa de detectar as placas de trânsito em vias públicas para a tomada decisão do veículo pode ser resolvida com a utilização de redes neurais convolucionais profundas.

II. TRABALHOS RELACIONADOS

Diversos trabalhos foram realizados na área de detecção e reconhecimento de placas de trânsito. Brkic [2] realiza um estudo das principais técnicas utilizadas para detecção. Segunda ela, há três categorias de métodos utilizados: baseados em cor, baseados em forma e baseados em aprendizado.

Malik e Siddiqi[3] utilizaram segmentação baseada em cor seguida por análise de forma. Através dessa técnica conseguiram detectar corretamente as placas de 169 imagens num total de 172, ou seja, 98.25% de acerto.

Brkic et. al. [4] utilizaram o detector de Viola e Jones [5]. Eles dividiram o teste em dois conjuntos de imagens. Em ambos conseguiram uma taxa de acerto maior que 90%.

Houben et. al. [6] compararam alguns métodos de detecção. Eles dividiram os dados em três categorias: placas proibitivas, placas obrigatórias e placas de perigo. Além do detector de Viola e Jones, usaram uma técnica chamada de Histograma Orientado a Gradiente (HOG) auxiliada por Análise Linear Discriminante (LDA) e uma terceira técnica chamada método baseado em modelo. Nas placas proibitivas, o detector de Viola e Jones teve um acerto de 98.8%. O método HOG com LDA acertou 91.3% nas placas proibitivas e 90.7% nas placas de perigo. Para os demais casos, todos os métodos apresentaram uma taxa de acerto menor que 75%.

Para a tarefa de reconhecimento de placas, tem-se conseguido bons resultados através do uso de Redes Neurais Convolucionais (CNN). Sermanet e LeCun[7] aplicaram esse método e conseguiram acurácia superior a 98%.

O mesmo método pode ser utilizado para se conseguir bons resultados em detecção e localização de objetos, como demonstrado por Ohn-Bar e Trivedi [8].

III. METODOLOGIA

Esse trabalho utiliza a base de dados GTSDB, disponibilizado pelo site do *Institut für Neuroinformatik*, que compreende 600 imagens de treino e 300 imagens de teste. Por ser uma base relativamente pequena para um treinamento de qualidade, e também com o intuito de realizar um treinamento adequado e que facilite o processo de avaliação do desempenho da rede, foi utilizado um método de fatiamento dessas imagens, que resultam em *patchs* ou fragmentos de imagem, das quais podem variar entre os grupamentos "com placa" ou "sem placa".

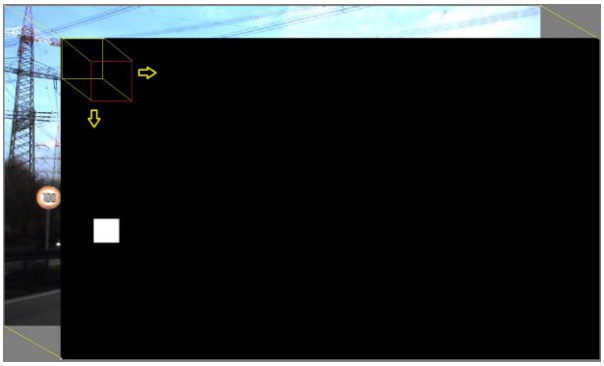
A. Pré-processamento dos dados

O resultado deste fatiamento resultaria em uma amostragem desproporcional de classes, visto que o número de imagens "sem placa" fatiadas seria muito maior do que o número de imagens com placa. Deste modo, buscou-se uma forma de contornar essa desproporcionalidade de dados, realizando diferentes formas de fatiamento das imagens.

As imagens de treino são fatiadas em diferentes escalas de *crops* e o processo utiliza dois métodos diferentes para o fatiamento, sendo que o fatiamento da região da imagem sem placa difere do fatiamento da região onde existe placa, e estes distintos processos são detalhados abaixo.

A imagem original é mapeada para uma imagem auxiliar binária, que indica, através dos coordenadas definidas pelo dataset, a localização das placas.

Figure 1. Processo de fatiamento da imagem.



Se o *kernel*, representado por uma matriz de dimensão 32x32 pixels, está sobre uma porção totalmente preta, significa que neste local correspondente à imagem, não estamos capturando um fragmento com placa da imagem.

Ao final, guardamos esta porção de imagem, redimensionada para 64x64 pixels, como parte do grupo de imagens da classe "sem placa". Esse processo é repetido deslocando-se o kernel pelo tamanho da sua dimensão horizontal e vertical, até que toda a imagem seja varrida, como visto na figura acima.

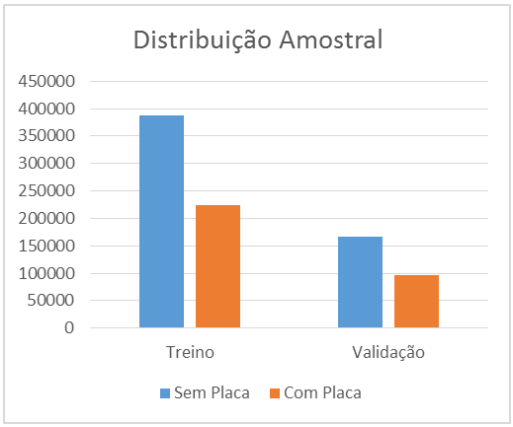
Por termos um dataset com 600 imagens e com dimensões de 1360x800 pixels, este processo resultaria em 553185 imagens pertencentes somente à classe "sem placa". Para reduzir este número, que traria alto custo computacional, deslocamos o kernel na imagem realizando saltos da ordem de 4 vezes o tamanho de sua dimensão horizontal e vertical. Com isso houve uma redução de amostras sem placa, que tornou possível a realização de todos os treinamentos, resultando em 64622 imagens de treino + 25072 images de validação, num total de 89694 imagens pertencentes à classe "sem placa", para a etapa de treinamento.

As regiões onde existe placa de trânsito são fatiadas em diferentes escalas, deslocando-se de 2 em 2 pixels, da esquerda para direita e do topo para o piso, onde a janela de captura, denominado *kernel*, é centralizada em relação ao ponto central da localização da placa. Estas coordenadas constam em arquivo fornecido pelo dataset utilizado.

O processo de fatiamento da imagem na região contendo placa é necessário para obtermos uma amostragem maior dos dados. Além disso, os *kernels* utilizados para capturar essa classe de imagens variaram em diferentes escalas, onde foram definidas as escalas de 16x16, 32x32, 64x64, 96x96 e 128x128 *pixels*, visando obter um grupamento de imagens com placas apresentando o menor ruído possível. Um *kernel* de escala *x* é utilizado apenas se o tamanho do mesmo for 40% maior que 70% da área da placa e ao mesmo tempo maior ou igual ao tamanho da área da placa, ou, se a área do *kernel* for maior ou igual a área da placa e a área do *kernel* for menor que 40% da área da placa. Estas condições visam um resultado que melhor represente uma amostra com placa. Ao término de cada fatiamento, a nova imagem contendo placa é redimensionada para 64x64 *pixels*. Isto é necessário pois a rede neural deve ser treinada com um padrão de dimensões.

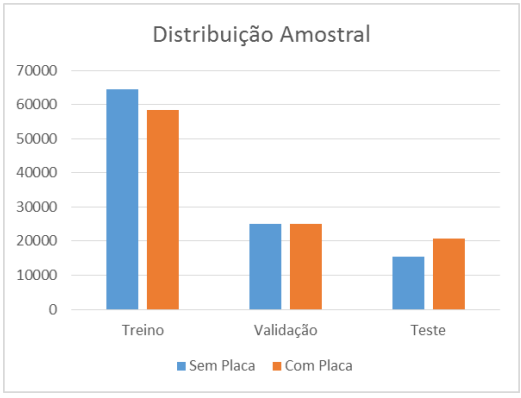
A distribuição amostral de imagens antes do método de fatiamento mais sofisticado é desproporcional e inviável devido ao alto volume de dados, como é visto no gráfico.

Figure 2. Distribuição amostral inicial.



Distribuição amostral dos grupos de imagens depois de realizar o método de fatiamento mais sofisticado e já constando as imagens de teste.

Figure 3. Distribuição amostral inicial.



B. Treinamento da rede

A rede utilizada AlexNet é uma rede neural convolucional profunda, utilizando uma taxa de aprendizado 0.01 inicial para as redes sem pré-treino e 0.001 para as rede com pré-treino. Todos os exeprimentos

C. *Pós-processamento dos dados*

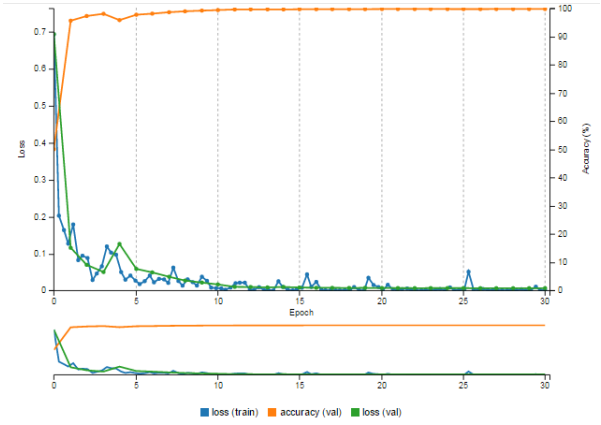
No pós-processamento é realizado uma modificação, via script, nas últimas camadas da rede neural, para que ao invés de obtermos apenas as saídas "com placa" e "sem placa", conseguirmos também um mapa probabilístico da localização da placa através de convoluções com a passagem da rede ao longo da imagem original, e nos casos de detecção, a imagem original é atualizada com um *bounding box* ou retângulo delimitador ao redor da área da placa.

IV. EXPERIMENTOS E RESULTADOS

Foram realizados 20 treinamentos, 10 com pré-treino e 10 sem pré-treino, sempre preservando a ordem de apresentação dos dados para o treinamento das redes bem como das demais características, tendo todos os experimentos com a mesma semente de aleatoriedade. A única característica que se modificou foi a taxa de aprendizado inicial, que na rede sem pré-treino foi de 0.01 inicial e para a rede com pré-treino, 0.001.

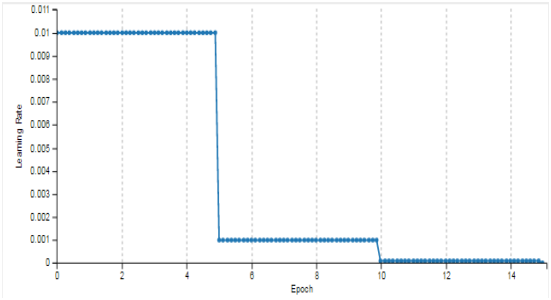
O número de épocas de treinamento foi inicialmente testado com 30, no entanto, com 5 épocas já é possível verificar a convergência da rede, e por isso o número foi reduzido para 15 épocas, a fim de reduzir o custo computacional dos testes.

Figure 4. Gráfico de convergência da rede neural com 30 épocas



A cada 5 épocas a taxa de aprendizado foi reduzida como mostra o gráfico abaixo.

Figure 5. Gráfico da taxa de aprendizado da rede neural



Para obter a acurácia de cada teste, foi realizada a passagem de todas as imagens fatiadas para a rede neural que realizou a classificação de cada uma detectando ou não, a presença de uma placa de trânsito.

Após a realização dos testes sob as mesmas condições para todas as outras redes, obtivemos os resultados abaixo.

A. *Redes sem pré-treino*

Para as redes sem o pré-treinamento houve um alto grau de acerto em imagens sem placa, no entanto, o rendimento caiu ao detectar imagens com placa.

TABLE I. ACURÁCIA DAS REDES SEM PRÉ-TREINO

Nome	**Acurácia**	
	Sem Placa	***Com placa***
R1	98,44%	94,62%
R2	98,45%	93,05%
R3	98,26%	98,85%
R4	98,47%	91,43%
R5	98,43%	92,93%
R6	98,33%	93,37%
R7	98,34%	92,60%
R8	98,40%	94,40%
R9	98,47%	92,56%
R10	98,55%	93,26%
Média	98,41%	93,11%
Média Total		95,76%
Desvio Padrão		2,72%

Compilando a matriz de confusão média dos 10 treinamentos sem pré-treino, foi obtida a matriz de confusão onde constam o número médio de acertos e erros das redes.

TABLE II. MATRIZ DE CONFUSÃO DAS REDES SEM PRÉ-TREINO

Real	**Predito**	
	Sem	***Com***
Sem	15139	244
Com	1430	19320

B. *Redes com pré-treino*

Para as redes com o pré-treinamento houve um alto grau de acerto em imagens sem placa, e quando comparado com a acurácia na detecção desta classe, vemos que as redes sem pré-treinamento se saem ligeiramente melhor, no entanto, quando verificamos o rendimento ao detectar imagens com placa, verificamos que a taxa de acerto se mantém em equilíbrio.

TABLE III. ACURÁCIA DAS REDES COM PRÉ-TREINO

Nome	Acurácia	
	Sem Placa	***Com placa***
R1	97,73%	97,97%
R2	97,68%	97,75%
R3	97,70%	97,63%
R4	97,74%	97,57%
R5	97,69%	97,57%
R6	97,72%	97,57%
R7	97,71%	97,89%
R8	97,69%	97,56%
R9	97,71%	97,78%
R10	97,79%	97,74%
Média	97,72%	97,70%
Média Total		97,71%
Desvio Padrão		0,1%

A matriz de confusão média dos 10 treinamentos com pré-treino resultou na seguinte matriz de confusão, onde constam o número médio de acertos e erros das redes.

TABLE IV. MATRIZ DE CONFUSÃO DAS REDES COM PRÉ-TREINO

Real	Predito	
	Sem	***Com***
Sem	15031	375
Com	453	20273

C. Imagem original pós-processada

Com um script capaz de varrer a imagem original, fazendo sucessivas convoluções da mesma para uma imagem auxiliar, conseguimos também um mapa probabilístico da localização da placa. No entanto, devido às convoluções, o mapa é gerado em pequena escala, mais precisamente, na escala de 24x42 *pixels.* Ao reajustar o mapa para o tamanho da imagem original, obtemos as respectivas coordenadas da área onde houve a detecção da placa e com estas coordenadas podemos traçar um *bounding box* ou retângulo delimitador ao redor desta área.

Figure 6. Mapa de probabilidade da localização da placa

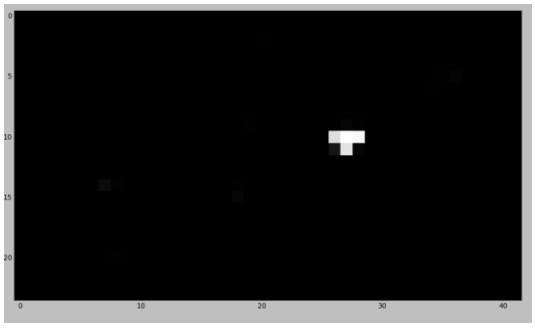


Figure 7. Detecção e localização da placa na imagem.

V. CONCLUSÃO

A quantidade amostral de imagens permitiu analisar além do desempenho da rede pré-treinada, investigar também se uma rede treinada do zero poderia obter melhores resultados, o que não foi constatado nesta pesquisa, visto que a acurácia média das redes pré-treinadas foi superior à média da acurácia das redes sem o pré-treino.

Por conta de sucessivas convoluções, o mapa de probabilidades de ocorrência de placa acaba sendo gerado em pequena escala. Ao ajustar o mapa probabilístico para a espacialidade da imagem original, sofremos uma perda de qualidade na localização da placa, pois ao serem escalonadas, as coordenadas acabam saindo da localização exata. Este problema deverá ser sanado em trabalhos futuros com a implementação de técnicas mais sofisticadas de convolução, para melhores resultados.

REFERÊNCIAS

[1] E. Ohn-Bar and M. M. Trivedi. "Multi-Scale Volumes for Deep Object Detection and Localization." Pattern Recognition, 2016.

[2] K. Brkić, "An overview of traffic sign detection methods." *Department of Electronics, Microelectronics, Computer and Intelligent Systems Faculty of Electrical Engineering and Computing Unska 3*, 2010.

[3] Z. Malik and I. Siddiqi, "Detection and Recognition of Traffic Signs from Road Scene Images," *Frontiers of Information Technology (FIT), 2014 12th International Conference on*, Islamabad, 2014, pp. 330-335

[4] K. Brkić, A. Pinz and S. Šegvić. "Traffic sign detection as a component of an automated traffic infrastructure inventory system." 33rd annual Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR). 2009.

[5] Paul Viola and Michael Jones. "Robust real-time object detection." In International Journal of Computer Vision, 2001.

[6] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing and C. Igel, "Detection of traffic signs in real-world images: The German traffic sign detection benchmark," *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Dallas, TX, 2013, pp. 1-8.

[7] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In Proceedings of International Joint Conference on Neural Networks (IJCNN'11), 2011.