
Sistema de detecção e identificação de placas de trânsito brasileiras utilizando técnicas de processamento digital de imagens e RCNN

Automatic detection and identification system of traffic signs using digital image processing techniques and R-CNN

Jozias Parente de Oliveira^{1*}, James Franklin Pereira Monteiro¹, André Luiz Printes¹, Israel Gondres Torné¹, Fábio de Sousa Cardoso¹, Angilberto Muniz Ferreira Sobrinho¹

RESUMO

Os Sistemas Avançados de Assistência ao Motorista (ADAS) são sistemas que buscam auxiliar o motorista na atividade da direção veicular, destacando-se pela capacidade em efetuar o mapeamento do ambiente, identificando situações de risco. Uma de suas aplicações consiste na detecção e identificação de placas de sinalização de trânsito. Para que esse processo seja feito com precisão, aplicam-se conceitos de visão computacional, processamento digital de imagens (PDI) e Redes Neurais. Desta forma, este trabalho propõe o desenvolvimento de um sistema de detecção e identificação automática de placas brasileiras de sinalização de trânsito, utilizando as técnicas previamente descritas. O modelo é composto por um sistema de RCNN – (*Region Based Convolutional Neural Networks*), construído em duas etapas. A primeira, utiliza de técnicas de PDI e classificador SVM para efetuar a extração da zona de interesse e categorização, obtendo uma taxa de precisão de 67,2 %, com bases de *benchmark*. A segunda, consiste em efetuar a classificação dos sinais obtidos anteriormente, através de um modelo de Rede Neural Convolutiva, com taxa de precisão de 99,26 %, com bases de *benchmark*, e para base de dados de placas brasileiras, 85,29 %.

Palavras-chave: Detecção e identificação de placas de trânsito; RCNN; Processamento Digital de Imagens.

ABSTRACT

The Advanced Driver Assistance Systems (ADAS) are systems that seek to help the driver to carry out the activity of vehicular steering, which highlights its ability to map the surrounding environment, identifying risk situations. One of the applications of this functionality is the detection and identification of traffic signs. For this process to be carried out with precision, it is necessary to apply concepts of computer vision, digital image processing (DIP) and Neural Networks. Thus, this work proposes the development of a system for the automatic detection and identification of Brazilian traffic signs, using the techniques previously described. The model is composed of an RCNN - (*Region Based Convolutional Neural Networks*) system, built in detection and classification steps. The first one uses DIP and SVM classifier techniques to extract the zone of interest and categorize it, obtaining an accuracy rate of 67.2%, in a test with benchmark bases. The second consists of classifying the signals obtained in the previous step, through a Convolutional Neural Network model, whose accuracy rate was 99.26%, in a test with benchmark bases, and for a database of Brazilian plates, 85.29%.

¹ Universidade do Estado do Amazonas.

*E-mail: jpoliveira@uea.edu.br

Keywords: Traffic Sign Detection and Classification; RCNN; Digital Image Processing.

INTRODUÇÃO

O Sistema Avançado de Assistência ao Motorista (ADAS) possui o intuito de auxiliar o motorista, tornando o ato de dirigir mais seguro, desta forma este sistema visa reduzir a alarmante estatística de 1,2 milhões de mortes no trânsito globalmente, conforme (WHO, 2011), buscando minimizar os erros humanos na condução veicular que, por sua vez, corresponde a 94% da causa de incidentes ocorridos, segundo Singh (2015).

Para alcançar tal objetivo de aprimorar a segurança veicular, este sistema utiliza de ferramentas que monitoram e analisam o ambiente ao redor do veículo através da aquisição e processamento digital de imagens, podendo desta forma realizar tarefas como: alerta de colisão e condições da via, detecção de pedestres e objetivos, entre outros. Além disso, o sistema visa minimizar os problemas causados pela excessiva e constante carga de informações ao motorista, especificamente em condições não favoráveis, o que torna o ato da direção veicular mais perigoso.

Desde sua concepção o ADAS vem evoluindo juntamente com a sua aplicação em mapeamento do ambiente, desta forma, abrangendo uma gama de objetos a serem devidamente detectados e identificados, assim como melhora na taxa de precisão e velocidade para reconhecimento dos objetos próximos ao veículo.

Tendo em vista este contexto, propõe-se a criação de um sistema de detecção e identificação de forma automática de placas de trânsito brasileiras aplicado ao sistema de auxílio à direção veicular. Para atingir tal objetivo, partisse da aplicação dos conceitos de processamento digital de imagem para efetuar as etapas de aquisição de imagens e tratamento; que consiste na digitalização, conversão para escala de cinza, limiarização e segmentação da imagem em análise. Posteriormente, as regiões de interesse são extraídas das imagens em análise e direcionadas para um modelo de *deep learning*, RCNN – (*Region Based Convolutional Neural Networks*), que é constituído de um classificador SVM e um modelo de Redes Neurais Convolucionais, onde o primeiro é responsável por fazer a catalogação das regiões de interesse, detectando se eles realmente contém as

características esperadas para uma placa de sinalização de trânsito de acordo com a forma, o segundo é responsável por efetuar a corretada identificação da região de interesse, informando com precisão qual é a sinalização correspondente ao sinal em análise.

Buscando uma apresentação de forma clara e objetiva, a pesquisa está ordenada nos seguintes tópicos:

Referencial - Expõe os principais conceitos sobre o ADAS, modelos e distinções sobre as placas de trânsito brasileiras, a problemática sobre a legibilidade das indicações de sinalização, assim como sobre os conceitos de imagem, o processamento digital da mesma e tomada de decisão. Aplicação de visão computacional para a resolução de problemas envolvendo detecção e classificação de objetos.

Métodos - Consiste no capítulo que exemplifica os processos de desenvolvimento adotados para este trabalho, assim como, as etapas de testes a serem aplicadas. Exemplificando e descrevendo as ferramentas, plataformas e conjuntos de dados utilizados. Assim como descreve a aplicação de cada uma das etapas descritas como métodos adotados.

Análise e Resultados - Expõe os resultados obtidos a partir da implantação da metodologia proposta, servindo como parâmetro para comparação da eficiência do modelo proposto no trabalho em relação aos *benchmarks* tradicionais, desta forma, podendo-se aferir uma conclusão, que será apresentada no fim do trabalho.

REFERENCIAL

Sistemas Avançados De Assistência Ao Motorista – ADAS

Sistemas avançados de assistência ao motorista (*Advanced Driver Assistance system* - ADAS) representa um conjunto de sistemas e subsistemas, que podem ser inteligentes e capazes de auxiliar o condutor, de forma passiva ou ativa. Segundo Brookhuis (2001), o conceito do ADAS inclui dentre diversas funcionalidades a detecção de “pontos cegos”, controle de cruzeiro adaptativo, controle de cruzeiro inteligente automatizado, condução de pelotão, entre outros. Além disso, segundo Reif (2014), o sistema ADAS tem principal objetivo fazer com que o automóvel efetue um mapeamento do ambiente ao redor, identificando situações de risco para o condutor, passageiros ou pedestres, devendo atuar de forma efetiva ou alertar sobre tais condições. Desta forma, o ADAS acarreta benefícios que visam reduzir ou até mesmo eliminar os erros do condutor,

tornando o ato da condução do veículo mais eficiente e segura, minimizando os problemas oriundos do clima, visão e ambiente.

Sobre o ADAS, pode-se ainda distinguir seus modos de operação, segundo Winner (2015), dos modos de operação do ADAS, destaca-se neste trabalho aquele onde o sistema provê informações e funções de aviso sobre situações externas, não havendo interferência direta no controle do automóvel, como as aplicações de detecção e reconhecimento de placas de sinalização de trânsito, aviso sonoro de proximidade, dentre outros.

Placas Brasileiras De Sinalização De Trânsito

Segundo o Código de Trânsito Brasileiro, as sinalizações de trânsito são definidas como:

Sinais de trânsito são elementos de sinalização viária que se utilizam de placas, marca viárias, equipamentos de controle luminosos, dispositivos auxiliares, apitos e gestos, destinados exclusivamente a ordenar ou dirigir o trânsito dos veículos e pedestres (DENATRAN, 2008)

O Conselho Nacional de Trânsito (CONTRAN) segmenta os padrões de sinalização brasileiro em seis categorias distintas, no entanto, de acordo com o tema proposto por este trabalho e sua respectiva delimitação de desenvolvimento, torna-se enfático a necessidade de expor os conceitos referentes às categorias de sinalizações verticais de regulamentação e de advertência, pois estas serão abordadas e utilizadas em específico nas metodologias aplicadas. A primeira, tem por finalidade transmitir aos usuários as condições, proibições, obrigações ou restrições no uso das vias urbanas e rurais, possuindo apenas as cores vermelho, branco e preto dispostas nas placas (CONTRAN, 2005). A segunda, tem por finalidade alertar aos usuários as condições potencialmente perigosas, obstáculos ou restrições na via ou adjacentes, possuindo apenas as cores amarelo ou laranja e preto dispostas nas placas (CONTRAN, 2005).

Conceito De Imagem

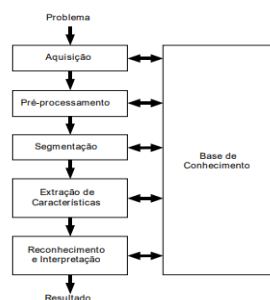
Toda imagem pode ser descrita como uma função $f(x,y)$, onde as variáveis x e y representam a localização de um ponto qualquer na imagem, e f representa a intensidade luminosa no determinado ponto, ou seja, pode-se afirmar que uma imagem é a representação de uma matriz de intensidade luminosa, capaz de ser equacionada numa função $f(x,y)$. Segundo Guimarães (2008), uma imagem digital $f[m,n]$ descrita em uma

espaço discreto 2D é derivada de uma imagem analógica $f(x,y)$ de um espaço contínuo 2D através de um processo de amostragem que é frequentemente referido como digitalização. Sobre os pontos de intensidade, pode-se afirmar que existem apenas três tipos de imagens: binária, monocromática e em cores.

Técnicas de Processamento Digital de Imagens

Sobre as etapas e técnicas que envolvem o sistema de aquisição e processamento de imagem, conforme Filho e Neto (1999), elas envolvem os procedimentos e sequência que estão representados e dispostos na Figura 1.

Figura 1 - Sistema de processamento de imagens



Fonte: Filho e Neto (1999)

Primeiramente é necessário haver o domínio do problema, que no projeto em questão seriam imagens que proporcionem realizar a detecção e identificação de placas de sinalização brasileiras de trânsito. Alinhado ao conhecimento do problema proposto, nota-se que todas as etapas subsequentes estão relacionadas a uma base de conhecimento, que consiste no local onde são armazenados o conhecimento já existente sobre o problema a ser resolvido, ou minimamente as informações necessárias para a resolução da problemática, contendo banco de dados catalogado e dividido em conjuntos de treino e teste.

A primeira etapa do processo refere-se à aquisição de imagens, é necessário a utilização de um sensor e um digitalizador. Onde o sensor converterá a informação óptica em sinal elétrico e o digitalizador transformará a imagem analógica em uma imagem digital.

A segunda etapa visa aperfeiçoar a qualidade da imagem para as etapas seguintes, tendo em vista que o objeto de interesse disposto pode apresentar pixels ruidosos, contrastes e brilho inadequado, entre outras imperfeições. Para que as irregularidades

sejam sanadas, esta fase é subdividida nos seguintes tópicos: aprimoramento, restauração, processamento de imagem em cores (mudança de formato de cores), compressão.

A terceira etapa, segmentação, é aquela onde uma imagem é dividida em diversas unidades, de forma significativa, ou seja, nos objetos de interesse que a compõem.

A quarta etapa refere-se a busca de extração das características das imagens resultantes, através de descritores que permitam caracterizar com precisão e com bom poder de discriminação entre os objetos destacados na imagem. A saída resultante é um conjunto de dados correspondentes àquela imagem, ou seja, uma sequência de vetores que contém as informações descritivas.

Por fim, a etapa de reconhecimento e interpretação. Segundo Filho e Neto (1999), é aquela onde é atribuído um rótulo a um objeto baseado em suas características, traduzidas por seus descritores. A tarefa de interpretação, consiste em atribuir um significado a um conjunto de objetos reconhecidos.

Para a resolução do problema proposto neste trabalho, nota-se que a quarta etapa resulta na detecção dos pontos de interesse do modelo e a quinta etapa consiste em realizar a interpretação e identificação de qual sinalização aquele objeto faz referência. Para que tais tópicos fossem implementados, adotou a aplicação dos conceitos de redes neurais e *deep learning*.

Redes Neurais

Segundo Russel e Novig (2003), redes neurais artificiais (RNA), são baseadas no conceito simplificado do funcionamento biológico do cérebro, onde um neurônio possui a função de processar e disseminar sinais elétricos.

De forma simplificada, pode-se afirmar que o conhecimento de uma RNA está conectado a sua estrutura de rede, onde tem-se em evidências as conexões, ou sinapses, entre as unidades, ou neurônios, que a compõem. É de importante valia, ressaltar que é atribuído um peso sináptico, valor numérico, a cada conexão, o que por sua vez caracteriza a força dela. O processo de aprendizagem de uma RNA consiste na adaptação dos seus pesos sinápticos, o que leva a criação de um caminho com uma resposta ao problema proposto embasada no peso das conexões estabelecida. Desta forma, o aprendizado ocorre de forma gradual e iterada, onde os pesos são modificados por diversas vezes, conforme os aprendizados já obtidos, há três grandes classes de métodos

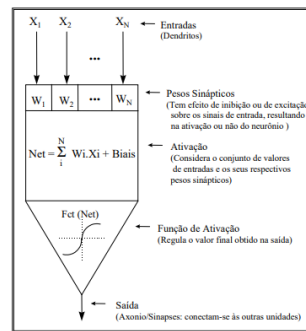
de aprendizado neural: aprendizado supervisionado, aprendizado semi-supervisionado e aprendizado não supervisionado (OSÓRIO e BITTENCOURT, 2000). Quanto a aplicação, podem ser utilizadas em diferentes tipos de tarefas, tais como: o reconhecimento de padrões, classificações, transformações de dados, previsões controle de processos e aproximações de funções.

Segundo Haykin (2009), a primeira etapa para o desenvolvimento de uma RNA é a definição da estrutura que será adotada pela mesma, determinando como será a atribuição dos neurônios utilizados, assim como a quantidade de camadas e conexões. Dentre as possibilidades de estruturas que podem ser adotadas, será utilizada a abordagem da Perceptron Multicamadas.

A Perceptron Multicamadas (*MultiLayer Perceptron* – MLP) apresenta uma estrutura de conjuntos virtuais de unidade sensoriais na camada de entrada, com uma ou mais camadas de neurônios ocultas e uma camada de saída. Sendo estas, redes inteiramente conectadas, de tal forma que cada um dos neurônios que a compõe uma camada conecta-se com todos da camada anterior. Os dados que são inseridos na estrutura de MLP compreendem informações características do problema em estudo, podendo ser relacionados a forma geométrica ou cor do objeto (HAYKIN, 2009).

Dos algoritmos que utilizam MLP, o *back-propagation* tem destaque, pois possui uma boa eficiência, realizando o treinamento da RNA e sendo aplicável em diversos problemas complexos. Este modelo baseia-se no aprendizado por tentativa e erro, usando um conjunto de dados de entrada e saída já identificados. Desta forma, é possível efetuar a correção de erro, com isso o modelo passa a aprender baseado em erros anteriores, uma vez que os pesos de cada conexão da RNA são iniciados de forma aleatória, o que acarreta num volume de erro muito grande, no entanto, a cada repetição é calculada os valores de SSE (*Sum Of Squared*) - Soma dos Quadrados dos Erros e MSE (*Mean Squared Error*) – Erro Quadrático Médio, o que permite ajustar a atribuição de pesos de cada conexão, desde a camada de saída até a entrada, fazendo com que na próxima iteração haja um resultado mais próximo do esperado para o treinamento.

Figura 2 - Exemplo de Neurônio Artificial



Fonte: Osório e Bittencourt (2000)

SVM – Support Vector Machine

Uma das técnicas de aprendizagem de máquina supervisionado mais difundidas é conhecida como SVM – *Support Vector Machine*. Segundo Akande (2014), este método tornou-se bastante popular pois alinha uma grande capacidade de generalização para soluções dos problemas relacionados a classificação de padrões com uma ótima eficiência.

Segundo Hsu, Chang e Lin (2003), o objetivo central do SVM é realizar a criação de um modelo que possa realizar a predição de a qual classe de um determinado dado a partir dos seus atributos, desta forma, o modelo de SVM busca traçar hiperplanos dividindo os grupos a partir de suas características.

Deep Learning e Reconhecimento de Imagens

Segundo Grace et al. (2018), uma das áreas de pesquisa mais desenvolvidas atualmente é a que abrange os conceitos de *Deep Learning* (DL), ou Aprendizado Profundo, uma vez que estes modelos possuem uma vasta área de aplicação como reconhecimento de fala, reconhecimento de objetos, sinais, visão computacional e outros.

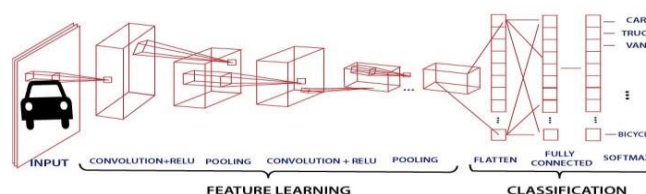
De forma geral, os processos de *deep learning* são semelhantes as redes neurais artificiais, porém efetuam uma verificação mais profunda acerca dos dados, pois utilizam um número mais significativo de camadas de neurônios, permitindo um aprendizado mais elaborado do modelo e a utilização de mais características da imagem em estudo para problemas de classificação. Segundo Oliveira (2018), a melhor técnica de *deep learning* buscando performance e custo computacional reduzido para reconhecimento de imagens

é utilização de *Neural Network Convolutions* (CNN), ou Rede Neural Convolucional (RNC).

Redes Neurais Convolucionais

As Redes Neurais Convolucionais utilizam um processo de convoluções sequenciais em um conjunto de dados, valendo-se da aplicação de filtros e métodos previamente aprendidos para obter as características desejáveis. Segundo Pacheco (2017), a grande vantagem desse método em relação a RNA tradicional quando aplicado em imagens, é que este possui o poder de efetuar a codificação das propriedades, fazendo com que o treinamento diminua sem afetar a precisão do modelo. De forma geral uma Rede Neural Convolucional tem a estrutura separada em uma etapa de Características de Aprendizagem e Classificação, como representado na Figura 3.

Figura 3 - Estrutura de uma Rede Neural Convolucional



Fonte: Doshi (2019)

A primeira etapa referente ao conjunto de características de aprendizagem consiste na camada de convolução. Esta camada efetua o processo de convolução, que possui a finalidade de criar um sinal resultante de duas funções ou sinais, através do produto deles. Segundo Goodfellow, Bengio e Courville (2016), a convolução é uma operação que efetua a transformação linear de uma matriz primária, podendo estar a ser um conjunto de 3 matrizes caso seja a representação de uma imagem RGB, de outra forma, para imagens em tons de cinza ou preta e branca, através da matriz primária é formada o kernel ou matriz de convolução. Em seguida, efetua-se a convolução propriamente dita, a multiplicação da matriz primária pela matriz de convolução, resultando numa terceira matriz que carrega as informações características da imagem ou matriz original.

A Camada de *Pooling* é responsável por reduzir as dimensões de uma imagem, porém, mantendo as informações mais relevantes. Desta forma, o *Pooling* gera uma saída resumida da imagem anteriormente filtrada, minimizando os custos computacionais. Segundo Oliveira (2018) a função de *Max Pooling* é a mais utilizada e indicada para

análise de imagens, pois este considera os valores máximo de cada região proporciona uma melhor visibilidade das áreas de alto contraste, que geralmente podem ser associadas a objetos e cores distintas em um ambiente.

A primeira etapa referente ao conjunto de classificação consiste na camada de *Flatten*. Esta é responsável por efetuar a adequação e redimensionamento da matriz resultante do processo anterior, para o padrão de entrada do modelo de classificação.

Na sequência, estão dispostas as camadas inteiramente conectadas. Estas são aquela em que cada neurônio está conectado a todas as saídas da camada anterior. Neste ponto todas as características adquiridas em outras camadas são utilizadas para encontrar padrões sobre a informação recebida na entrada, de forma tal que haja os ajustes de pesos para cada conexão.

A Camada de *SoftMax*, é a responsável por obter a probabilidade de a imagem de entrada ser de determinada classe, sendo assim, ocorre posteriormente a camada totalmente conectada.

Por fim, a camada de classificação. É a responsável por efetuar o cálculo de perdas de entropia de cada classe de saída, uma vez que esta é mutuamente exclusiva com o valor atribuído a classificação. Sendo assim, essa camada é posiciona de tal forma que receba os valores oriundas da camada de *SoftMax*, efetuando o cálculo de perda correspondente a cada classe de saída.

Deteccção de Regiões de Interesse

Uma vez que uma RNC normalmente não consegue efetuar deteção e classificação em conjunto, foi desenvolvido o conceito de Rede Neurais Convolucionais baseadas em regiões.

R-CNN – Redes Neurais Convolucionais Baseadas em Regiões

Segundo Girshick (2014), as chamadas Redes Neurais Convolucionais Baseadas em Regiões ou R-CNN (*Region-based Convolutional Neural Network*) buscam segmentar uma imagem em sub-regiões, posteriormente cada região possui a dimensão alterada para ser aplicada a estrutura de CNN já exposta, por fim, a rede neural efetua a classificação da região.

Segundo Girshick (2014), as etapas de um algoritmo de R-CNN consistem em:

1. Realizar uma pesquisa seletiva para extrair várias propostas de regiões de alta qualidade da imagem de entrada, sendo estas selecionadas em escalas, formas e tamanhos. Cada região é rotulada com uma classe e delimitada por uma *boundingbox*. As regiões de interesse são obtidas através da aplicação do conceito de HOG;
2. É realizado a escolha de um CNN, pré-treinado, sendo os resultados truncados antes da camada de saída. Cada proposta de região é redimensionada para ser aceitável aos padrões da CNN, de forma que produza os recursos extraídos para a proposta de região;
3. As características extraídas e classe rotulada de cada região, são utilizadas no treinamento de máquinas de vetores de suporte – SVM, onde cada módulo deste determina de forma individual se um exemplo contém uma classe específica;
4. Por fim, cada característica e rótulos atribuídos as *boundingbox*, são aplicadas para o treinamento de um modelo de regressão linear, de forma a prever a classe da região selecionada.

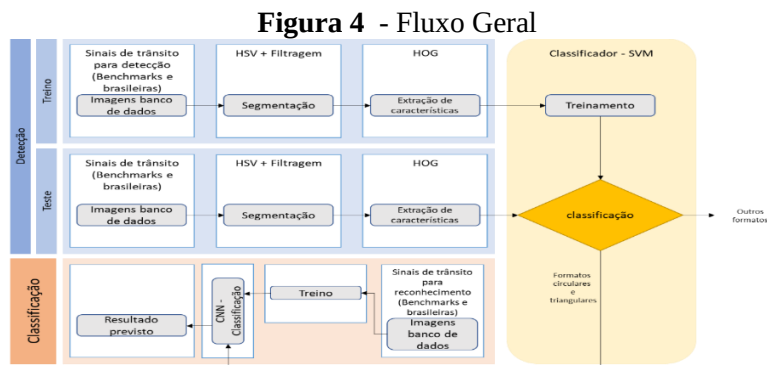
HOG – *Histogram of Oriented Gradients*

O HOG (*Histogram of Oriented Gradients*) ou Histograma da Orientação dos Gradientes, consiste num descritor de imagens utilizado principalmente para projetos de detecção de objetos, pois o resultado proveniente dos filtros de HOG apresenta as características da imagem em análise, podendo o resultado ser extraído e adicionado em um *pipeline* de aprendizado de máquina para a detecção da classe à qual as características ali encontradas pertencem, sendo amplamente utilizada pois possui uma larga capacidade de representar a imagem ou objeto em questão por um simples vetor.

Segundo Mallick (2016), o funcionamento do HOG consiste em coletar informações de importância significativa sobre a direção de gradiente de uma imagem dividida em diversas células. Efetuando este processo nas seguintes etapas: Cálculo do módulo e direção dos gradientes horizontais e verticais da imagem; Divisão da imagem em múltiplas células; Cálculo do histograma dos gradientes; Normalização das células e a concatenação das informações em vetor.

MÉTODOS

Seguindo os conceitos anteriores expostos no referencial, dividiu-se o processo proposto em quatro etapas de implementação, de forma mais específica: Banco de dados; Plataforma; Detecção; Classificação.



Fonte: Autoria própria (2021)

Bancos de Dados

Para o desenvolvimento desse trabalho, foram utilizadas bases de dados construídas com o propósito para situações de implementação em algoritmos de detecção e classificação de placas de sinalização de trânsito, possuindo divisões entre os dados referentes a treino e teste, assim como arquivos com características de interesse da imagem para o problema a ser solucionado, no caso, as coordenadas de *boundingbox* com a região de interesse (ROI), tamanho da mesma e categoria pertencente.

Para as questões que envolvem detecção dos objetos de interesse, foram utilizadas as seguintes bases no estudo: *German Traffic Sign Detection Benchmark* – GTSDb (HOUBEN, 2013) e *Belgium Traffic Sign Dataset* - BelgiumTSD (MATHIAS, 2013). Já para as questões que envolvem classificação dos objetos, foram utilizadas as seguintes bases: *German Traffic Sign Recognition Benchmark* – GTSRB (STALLKAMP, 2011) e Banco de Dados I (PEREIRA, 2018).

Devido à inexistência de um banco de dados utilizado como *benchmark* para detecção e classificação de placas de sinalização brasileiras, optou-se por utilizar, para fins de análises quantitativas das técnicas propostas neste trabalho, as bases de GTSRB, GTSDb e BelgiumTSD. Pois estas bases permitem a avaliação de desempenho e precisão do modelo de extração de ROIs e classificação. Sendo assim, os bancos de dados com placas de sinalização brasileiras, como o Banco de Dados I, foram utilizados para

exemplificar a possibilidade da aplicação do modelo de detecção e reconhecimento aqui proposto.

Plataforma

A plataforma IDE utilizada para o desenvolvimento dos algoritmos de detecção e classificação das placas de trânsito foi o *Google Colab*, consultando repositórios no *Google Drive*. Quanto a linguagem de programação adotada, utilizou-se Python, pela sua vasta aplicação e dependências visando tratamento de dados e aprendizado de máquina possuindo bibliotecas como *OpenCV*, *Numpy* e *SciPy*. Além do uso de *Scikit-Learn* e *TensorFlow*, para construções dos modelos de aprendizado de máquina.

Detecção

O processo de detecção representa as etapas de pré-processamento, tratamento e extração as regiões de interesse que serão recebidas pelo modelo classificador SVM.

A primeira etapa consiste em efetuar um pré-processamento nas imagens do banco de dados, de forma tal que seja possível distinguir as placas de sinalização de trânsito do ambiente ao redor e posteriormente selecionar apenas a região de interesse onde ela se encontra. A imagem de entrada em RGB é convertida em HSV (*Hue, Saturation, Value*), uma vez que o formato RGB é bastante sensível a questões de luminosidade, diferentemente do HSV.

A segunda etapa consiste na aplicação de filtros na imagem, onde são utilizadas as técnicas de limiar para detectar as cores das placas e sinalização de trânsito, sendo o procedimento mais importante referente a determinação do valor atribuído ao limiar. Em seguida, é efetuado um tratamento para eliminar os falsos candidatos que foram erroneamente detectados como placas de trânsito pela limiarização. Para isso é efetuado o processo de segmentação e filtragem, aplicando operações de erosão e dilatação a imagem segmentada.

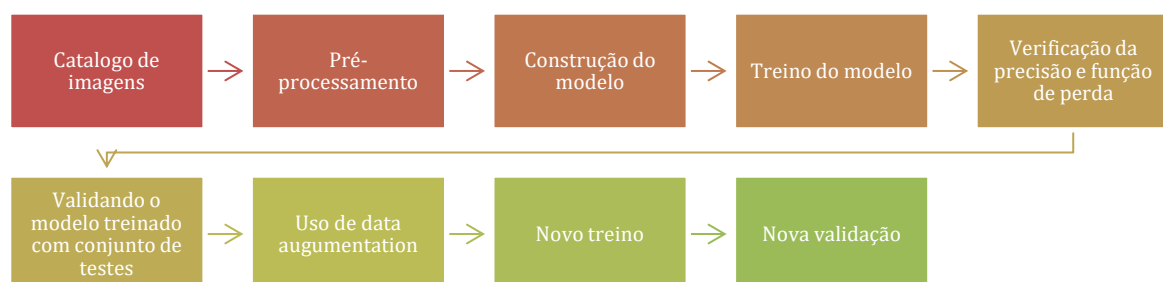
A terceira etapa consiste na extração de características das imagens através das regiões de interesse. Neste trabalho foi utilizado a técnica de HOG. Em seguidas essas características extraídas são direcionadas para o treinamento e subsequente teste, de uma camada de SVM.

A quarta etapa e última do processo de detecção refere-se à utilização da técnica de SVM, onde é focado o reconhecimento da forma geométrica obtidos no processo anterior de segmentação. Desta forma, uma vez que os vetores de características são calculados e informados através da HOGs, estes são usados no método de SVM. Este método é de interessante aplicação pois possui uma robustez a diversos fatores como translações, rotações e escala. O método SVM busca encontrar um hiperplano ótimo para separar de forma mais distinta possível as classes em questão, assumindo que são linearmente separáveis. Neste trabalho, utilizou-se um classificador SVM linear de forma que este diferencie o formato da placa de trânsito recebida entre: circular, triangular, outros.

Classificação – CNN

Após a etapa de identificação do formato da placa de sinalização, é utilizado um modelo de Redes Neurais Convolucionais para efetuar a classificação dos sinais, seguindo as etapas exemplificadas na Figura 5.

Figura 5 - Fluxo de implementação da CNN



Fonte: Autoria própria (2021)

A primeira etapa consiste em realizar a separação das imagens presentes no banco de dados em classes e em pastas distintas de treino e teste para o modelo. Desta forma, ele terá material para efetuar processos de treino e validação de um modelo de CNN, com dados suficientes para evitar ocorrências de *overfitting* e *underfitting*.

A segunda etapa consiste em realizar os pré-processamentos necessário nas imagens que serão utilizadas. Primeiramente, ocorre a conversão do formato RGB para HSV, seguido da equalização do histograma. Em segundo lugar, a imagem possui seu tamanho reajustado, inicialmente para 48x48, que é o tamanho adotado para a camada de

entrada do modelo. Em terceiro lugar, são obtidas as informações das classes correspondentes a cada imagem e vetorizadas para utilização posterior do modelo.

A terceira etapa consiste na construção do modelo propriamente dito, para isso, usou-se as bibliotecas *keras* e *tensorflow*. Sendo assim, utilizam-se das funções destas bibliotecas para a criação da estrutura da CNN. Neste ponto, foram construídos quatro modelos de CNN distintos, apresentando variabilidade em quantidade de camadas convolucionais, camadas totalmente conectadas, tipos e tamanho de filtros aplicados, de forma a submetê-los a treino com o conjunto de bases, de forma melhor resultado seja escolhido para aplicação final, em condições de teste. Os modelos propostos são apresentados na Tabela 1.

Alguns dos parâmetros adotados na construção do modelo: *InitialLearnRate* = 0,01; A função de SGD foi adotado para minimizar a função de perda, sob uma taxa de decaimento de 10^{-6} , momentum = 0,9. Além disso o parâmetro de *LerningRateScheduler* foi utilizado como exposto no item anterior. Aqui ainda foram iniciadas para observação as métricas de *accuracy*, ou precisão, do modelo e *loss function*, ou função de perda, sendo aqui utilizada a *categorical crossentropy*.

Quadro 1 - Modelos de CNN propostas

A	B	C	D
Camada de entrada			
Conv - 32	Conv - 32	Conv - 32	Conv - 64
ReLU			
MaxPool			
Conv - 32	Conv - 32	Conv - 32	Conv - 128
ReLU			
MaxPool			
Conv - 64	Conv - 32	Conv - 32	
ReLU			
MaxPool			
Conv - 64	Conv - 32		
ReLU			
MaxPool			
Conv - 128			
ReLU			
MaxPool			
Conv - 128			
ReLU			
MaxPool			
Completamente conectada			
ReLU			
Completamente conectada			
SoftMax			

A quarta etapa consiste em efetuar o treino, neste estágio, o conjunto de dados referente a treino é adicionado ao modelo, de forma tal que este passe a conhecer os dados que serão trabalhados, num certo nível de semelhança com os de teste, sendo assim, o modelo efetua os ajustes de pesos e melhora as conexões e predições a cada iteração com os dados de treino. Os resultados quanto a performance de cada modelo proposto é exposta no item de Análise e Resultados, apenas aquele com a melhor performance foi selecionado para continuar nas etapas subsequentes.

A quinta etapa consiste em verificar a precisão do modelo e os valores da função de perda para cada iteração. É esse estágio que se tem uma avaliação mais visual do progresso de aprendizagem do modelo de acordo com cada iteração feita por ele.

A sexta etapa consiste em avaliar o modelo treinado com base no conjunto de dados referente a teste. Neste estágio, o modelo já treinado é utilizado para predizer as classes do conjunto de testes e posteriormente ter a taxa de acerto avaliada.

A sétima etapa consiste na tentativa de melhorar o cenário de treino do modelo, de tal forma, que este possua um treinamento maior para ocasiões de aquisições diversas. Para tal, foi utilizada a função de *DataAugmentation* nos dados de treino, utilizando-se desta técnica para aumentar a quantidade de dados através da criação de cópias ligeiramente modificadas dos dados já presentes no conjunto de dados. Esta função tem a atribuição de agir como um regularizador, ajudando a reduzir o *overfitting* num treino do modelo.

A oitava etapa consiste em efetuar um novo treino do modelo, com a base de dados já tendo passado pelo processo de *Augmentation* na etapa anterior.

A nona etapa consiste em reavaliar o modelo com base na curva de precisão do modelo e valores da função de perda para cada iteração. Desde forma, verificando se houve melhora com a aplicação da técnica de *Augmentation*.

ANÁLISE E RESULTADOS

Resultados de Detecção

O modelo de detecção utilizando SVM, proposto na metodologia, foi testado utilizando a base de dados BelgiumTSD, para nível de benchmarks, devido ao grande

número de imagens e placas presentes na mesma, sendo que é possível a universalização e aplicação dele às placas brasileiras contanto que haja base de treinamento adequado, uma vez que as formas e cores entre o modelo brasileiro e o belga, apresentam similaridades.

Os resultados para ambientes de treino de teste no modelo SVM estão expostos na Tabela 1.

Tabela 1 - Resultados modelo de SVM

Modelo	Precisão
Treino	0,799
Teste	0,672

Fonte: Autoria própria (2021)

Segue um exemplo de saída proveniente do modelo de SVM, na Figura 6, com a imagem real apresentando plano de fundo e características de ambiente, utilizando de etapas de segmentação e técnicas de HOG para obtenção da região de interesse, resultando na imagem recortada apenas com a placa de trânsito na saída do modelo.

Figura 6 - Resultado de saída do modelo de SVM - exemplo



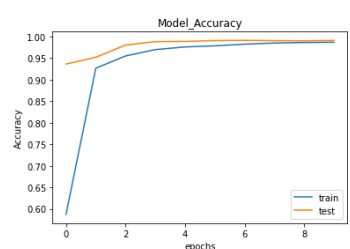
Fonte: Autoria própria (2021)

Resultados de Classificação

Redes Neurais Propostas

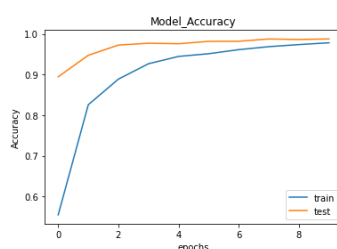
Nas Figuras 7 a 14 estão ilustrados os resultados das curvas de precisão e função de perda para cada modelo proposto na metodologia.

Figura 7 - Precisão RNC A



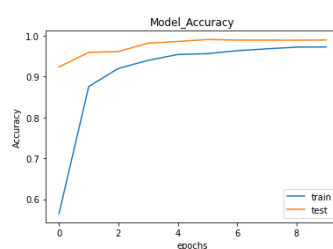
Fonte: Autoria própria
(2021)

Figura 8 - Precisão RNC B

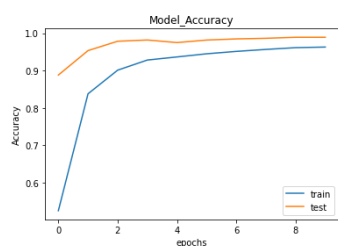


Fonte: Autoria própria
(2021)

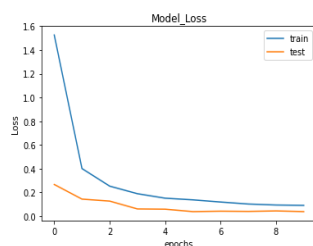
Figura 9 - Precisão RNC C



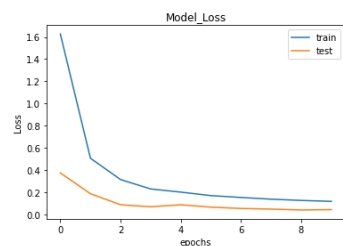
Fonte: Autoria própria
(2021)

Figura 10 - Precisão RNC D

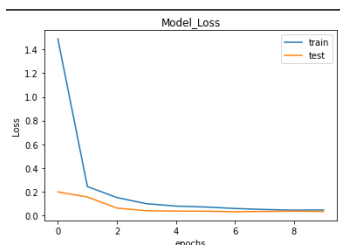
Fonte: Autoria própria
(2021)

Figura 12 - Função de perda - RNC B

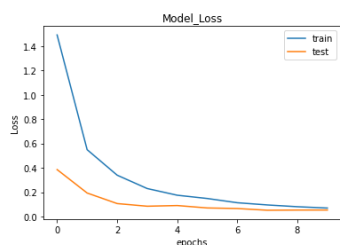
Fonte: Autoria própria
(2021)

Figura 14 - Função de perda - RNC D

Fonte: Autoria própria
(2021)

Figura 11 - Função de perda - RNC A

Fonte: Autoria própria
(2021)

Figura 13 - Função de perda - RNC C

Fonte: Autoria própria
(2021)

Pode-se analisar os resultados consolidados na Tabela 2, que apresenta as principais características dos diferentes modelos em ambiente de treinamento.

Tabela 2 - Resultados do treinamento dos modelos propostos

Rede	Accuracy	Loss	Val_accuracy	Val_loss	Tempo médio por epoch (s)
A	0,9870	0,0453	0,9911	0,0322	400
B	0,9724	0,0905	0,9894	0,0380	255
C	0,9780	0,0692	0,9875	0,0529	700
D	0,9631	0,1165	0,9892	0,0416	200

Fonte: Autoria própria (2021)

Observa-se que a Rede A, que possui mais camadas e aplicação de filtros, apresentou um valor de precisão próximo de 99%, em detrimento de um tempo de treinamento total de 1 hora e 6 minutos. Já a Rede B, com duas camadas convolução a menos que A, apresentou uma precisão de aproximadamente 97% num tempo total de 42 minutos. Por sua vez, a Rede C, com apenas duas camadas de convolução e aumento significativo na quantidade de filtros, apresentou uma precisão maior que C, no entanto, possuiu o maior tempo de treinamento, totalizando cerca de 1 hora e 56 minutos. Por fim,

a Rede D, apresentou o pior índice de precisão dos modelos propostos, no entanto, foi o mais rápido no período de treino, cerca de 33 minutos.

Levando em conta os valores e discussões apresentados anteriormente, optou-se por seguir com o modelo proposto na Rede A, considerando-o como melhor custo-benefício entre tempo de treino, precisão, aplicação de filtros e função de perda.

Aplicação da Rede Neural Escolhida

Ainda se utilizando do conjunto de dados GTSRB, obtiveram-se os seguintes resultados de classificação da CNN, mostrada na matriz de confusão presente na Tabela 3 (Performance geral de teste do modelo):

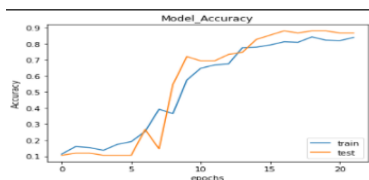
Tabela 3 - Resultado Rede A aplicada a conjunto de teste

Rede	Accuracy	Loss	Tempo médio (s)
A	0,9926	0,032	64

Fonte: Autoria própria (2021)

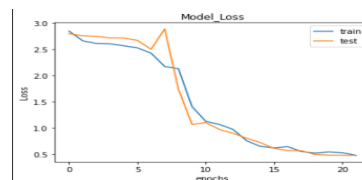
Para fim de demonstração de aplicação num conjunto de dados com placas brasileiras de trânsito, utilizou-se da base de dados desenvolvida por Pereira (2018), na UFSC. A Rede A proposta foi treinada com um valor de 22 iterações, apresentando a seguinte curva de precisão e de função de perda ilustradas nas Figuras 15 e 16.

Figura 15 - Precisão RNC A - conjunto de dados brasileiro



Fonte: Autoria própria (2021)

Figura 16 - Função de perda RNC A - conjunto de dados brasileiro



Fonte: Autoria própria (2021)

Quando aplicada aos dados com placas de trânsito brasileira, a Rede A apresentou a seguinte performance de treino, exposta da Tabela 4.

Tabela 4 - Resultado do treinamento em conjunto de dados BR

Rede	Accuracy	Loss	Val_accuracy	Val_loss	Tempo médio (s)
A	0,8389	0,4771	0,8667	0,4753	4

Fonte: Autoria própria (2021)

Utilizando-se o conjunto de dados referente a testes com placas brasileiras de trânsito, obtiveram-se os resultados de classificação da CNN conforme Tabela 5 (Performance geral de teste do modelo):

Tabela 5 - Resultados do modelo em conjunto de dados BR - teste

Rede	Accuracy	Loss	Tempo médio por epoch (s)
A	0,8529	0,4839	1

Fonte: Autoria própria (2021)

CONCLUSÕES

O trabalho em questão teve o intuito de desenvolver um sistema de detecção e identificação automática de placas brasileiras de sinalização de trânsito, aplicadas ao auxílio à direção veicular, de forma a utilizar técnicas de processamento digital de imagens e redes neurais.

Desta forma, foram apresentados os principais conceitos utilizados para o desenvolvimento da pesquisa e implantação do modelo proposto como: o ADAS e suas necessidades para melhoria da segurança dos motoristas; das placas de trânsito brasileiras segundo as normas regulamentadoras nacionais; assim como também foram apresentados os conceitos de imagem e as principais técnicas de processamento digital de imagem; os conceitos de redes neurais artificiais e SVM, assim como suas aplicações; conceito de *deep learning* e como as técnicas do mesmo podem ser utilizadas para reconhecimento de imagens e por fim, de técnicas de detecção e extração de objetos de interesse.

O modelo de RCNN proposto, seguiu do princípio da utilização de bases já estabelecidas para níveis de comparação com benchmarks, chegando a uma taxa de precisão de 99,26 % para base de GSTRB e para base de dados de placas brasileiras o valor de 85,29 %, sendo considerado satisfatório para o conjunto de dados e condições já exploradas no decorrer do trabalho. O modelo optado foi um RCNN, que por sua vez foi desenvolvido em duas etapas: a primeira focada em detecção de placas de trânsito, utilizando técnicas de processamento de imagem e modelo de SVM para efetuar a extração da zona de interesse, obtendo uma precisão da taxa de 67,2 % em ambiente de teste, para o conjunto de dados utilizado (BelgiumTSD). A segunda etapa consiste em utilizar o resultado proveniente do modelo de SVM para efetuar a classificação dos sinais de trânsito, seguindo um modelo de Rede Neural Convolucional clássico cuja precisão já foi abordada.

Sendo assim, avaliando os resultados obtidos e comparando os com os objetivos definidos para o projeto, pode-se considerá-los satisfatório, visto que foi possível: Desenvolver um modelo de detecção e classificação conjunta; Obter um resultado

comparável com os benchmarks tradicionais, validando a eficácia do modelo proposto; Utilizar dados de placas brasileiras de trânsito para treino de teste do modelo; Obter resultados satisfatórios de predição para placas de sinalização de trânsito brasileiras, uma vez que não há uma base de benchmark padrão para este tipo de conjunto de dados.

Por fim, acredita-se que esse trabalho pode ser expandido através de estudos direcionados a gerar uma base de dados obtidas em necessários de trânsito real, de forma que possa ser utilizada para futuros benchmarks, e efetuar a classificação dos dados presentes na mesma, a fim de que seja possível ter uma maior confiabilidade na aplicação do modelo em cenários de placas de trânsito brasileiras. Outra possibilidade, seria a adaptação do código e das técnicas aqui propostas para a criação de um sistema embarcado, que efetuar a detecção de forma automática e em tempo real, obtendo a informação de uma câmera e exibindo o resultado em uma tela.

REFERÊNCIAS

AKANDE, A. **Performance comparison of svm and ann in predicting compressive strength of concrete**. v. 16, p. 88-94, 2014.

BROOKHUIS, Karel A.; DE WAARD, Dick; JANSSEN, Wiel H. Behavioural impacts of Advanced Driver Assistance Systems—an overview. **European Journal of Transport and Infrastructure Research**, [S.l.], v. 1, n. 3, junho 2001. ISSN 1567-7141.

CONTRAN. **Manual Brasileiro de Sinalização de Trânsito – Volume 1 – Sinalização Vertical de Regulamentação**. 2005. Disponível em: http://vias-seguras.com/documentos/documentos_temas_o_a_z/doc_sinalizacao_e_seguranca_do_trnsito/manual_brasileiro_de_sinalizacao_de_transito_volume_i. Acesso em: 19/09/2020.

CONTRAN. **Manual Brasileiro de Sinalização de Trânsito – Volume 2 – Sinalização Vertical de Advertência**. 2007. Disponível em: <https://www.mobilize.org.br/midias/pesquisas/manual-brasileiro-de-sinalizacao-vol-ii.pdf>. Acesso em: 19/09/2020.

DENATRAN. **Conselho Nacional De Trânsito - Código de Trânsito Brasileiro**. 2008. Disponível em: <https://goo.gl/M2KkMC>. Acessado em 19/09/2020.

DOSHI, Sanket; **Traffic Sign Detection Convolutional Neural Network**. 2019. Disponível em: <https://towardsdatascience.com/traffic-sign-detection-using-convolutional-neural-network-660fb32fe90e>. Acesso em: 15/09/2021.

FILHO, Ogê Marques; NETO, Hugo Vieita; **Processamento Digital de Imagens**. Rio de Janeiro, Brasport, 1999. ISBN 8574520098.

GIRSHICK, R. et al. **Rich feature hierarchies for accurate object detection and semantic segmentation**. United States, 2014.

GOODFELLOW, I; BENGIO, Y; COURVILLE, A. **Deep Learning**: Adaptive Computation and Machine Learning series. Cambridge, MA: MIT Press: 2016.

GRACE, K., SALVATIER, J., DAFOE, A., ZHANG, B., EVANS, O. **When Will AI Exceed Human Performance?**. Evidence from AI Experts., 2018

GUIMARÃES, Marly. Amostragem e Quantização. **Digital Signal Processing**, Manaus, jun. 2008.

HAYKIN, S. **Neural networks and learning machines**, 3rd ed. Ontario, Canada: Pearson, 2009.

HOUBEN, Sebastian; STALLKAMP, Johannes; SALMEN, Jan; SCHLIPSING, Marc; IGEL, Christian. **Detection of traffic signs in real-world images: the german traffic sign detection benchmark**. INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN'13), 2013, Dallas. Proceedings. Piscataway: IEEE, 2013. p.1–8

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. **A practical guide to support vector classification**. v. 101, p. 1396-1400, 2003.

MALLICK, S. **Histogram of Oriented Gradients**. 2016. [Online; Acessado em 15 de dezembro de 2021]. Disponível em: <http://www.learnopencv.com/histogram-of-oriented-gradients>

MATHIAS, M. et al. **Traffic sign recognition—how far are we from the solution?** In: IEEE. Neural Networks (IJCNN), The 2013 International Joint Conference on. [S.l.], 2013. p. 1–8.

OLIVEIRA, Wellington. **Software para reconhecimento de espécies florestais a partir de imagens digitais de madeiras utilizando deep learning**. Universidade Tecnológica Federal do Parana, 2018.

OSÓRIO, Fernando; BITTENCOURT, João Ricardo; **Sistemas Inteligentes baseado em Redes Neurais Artificiais aplicados ao Processamento de Imagens**. 2000. Disponível em: https://www.researchgate.net/profile/Fernando_Osorio2/publication/228588719_Sistemas_Inteligentes_baseados_em_redes_neurais_artificiais_aplicados_ao_processamento_de_imagens/links/0912f51001cc71ad2b000000/Sistemas-Inteligentes-baseados-em-redes-neurais-artificiais-aplicados-ao-processamento-de-imagens.pdf. Acesso em: 15/09/2020.

PACHECO, A. **Redes neurais convolutivas**. Brasil, 2017.

PEREIRA, Daniel Bitencourt. **Análise e Implementação de Sistema Para Detecção de Placas Brasileiras de Sinalização de Trânsito**. Brasil, 2018

REIF, K. **Brakes, Brake Control and Driver Assistance Systems**. ViewegTeubner Verlag, 2014. ISBN 3658039779.

RUSSEL, S.; NOVIG, P. **Artificial Intelligence: A Modern Approach**. 2a. edição. Prentice Hall, 2003.

SINGH, S. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. **National Highway Traffic Safety Administration**, n. February, p. 12, 2015.

STALLKAMP, Johannes; SCHLIPSING, Marc; SALMEN, Jan. **The German Traffic Sign Recognition Benchmark: A multi-class classification competition**. Neural Networks (IJCNN). The 2011 International Joint Conference on. 2011

WHO. Global status report on road safety 2015. **WHO Library Cataloguing-in-Publication Data Global**, p. 340, 2015.

WINNER, H. et al. **Handbook of Driver Assistance Systems**. Springer-Verlag GmbH, 2015. 1602 p. ISBN 3319123513.

Recebido em: 28/02/2022

Aprovado em: 25/03/2022

Publicado em: 30/03/2022