

Classificação de Sinais de Trânsito para Direção Autônoma via Rede Neural Convolutacional

Nicomar Oliveira, Pedro Nuno Moura, Carlos A. V. Campos
Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Email: {nicomar.oliveira, pedro.moura, beto}@uniriotec.br

Resumo—Sinais de trânsito são elementos fundamentais que servem de alerta e orientação à direção autônoma de veículos. Nesse sentido, um dos principais problemas de pesquisa abordados na literatura relacionada é o de classificação automática de sinais de trânsito. Uma vez que veículos autônomos podem ser utilizados em diferentes países, onde os sinais de trânsito podem ter diferentes símbolos e significados, é necessário que os classificadores sejam capazes de se adaptar de acordo com o conjunto de sinais de cada país. Dado isso, este trabalho propõe o treinamento de modelos de aprendizado de máquina utilizando datasets públicos de placas de trânsito de diferentes países, com o objetivo de gerar classificadores suficientemente robustos para utilização nesses diversos países. Assim sendo, são apresentadas algumas estratégias para treinamento dos modelos de classificação, bem como uma avaliação experimental dessas diferentes estratégias, sendo destacada aquela que levou aos melhores resultados.

Palavras-chave: Sinais de Trânsito, Aprendizado Profundo, Classificação, Direção Autônoma, Aprendizado de Máquina.

I. INTRODUÇÃO

A Inteligência Artificial (IA) simula a inteligência humana através de mecanismos e/ou dispositivos. No passado, técnicas de IA eram vistas com desconfiança, mas hoje vêm se tornando cada vez mais presentes nas vidas das pessoas. O progresso nesta área vem sendo possível por alguns avanços específicos como o aumento da capacidade de processamento das máquinas e técnicas de multiprocessamento usando *Graphics Processing Unit* (GPU) que permitem multiplicação de matrizes em paralelo. Além disso, fabricantes de softwares desenvolveram bibliotecas de código aberto que simplificam a leitura, conversão, processamento dos dados e cálculo de métricas de desempenho, assim como o treinamento e execução dos modelos de classificação e predição.

Dentre as técnicas pertencentes à IA, existem algoritmos que buscam simular nas máquinas o processo de aprendizado do cérebro humano, subárea denominada de Aprendizado de Máquinas. Neste segmento, a partir desta última década, as redes neurais tornaram-se as principais componentes pelo resultados revolucionários obtidos, tais como em [1] e [2]. Essas redes nos permitem reconhecer imagens e classificá-las com alto grau de acurácia [3]. Dado isso, a motivação deste trabalho vem do fato de que estas redes têm despontado no reconhecimento de sinais de trânsito por veículos autônomos e a possibilidade de utilização destes veículos em diferentes países requer adaptações nas técnicas de treinamento, para que

os modelos gerados sejam capazes de se adaptar aos sistemas de sinalização de cada país.

A direção autônoma divide-se em 6 níveis de autonomia, de acordo com a Sociedade de Engenheiros da Mobilidade (SAE), variando do Nível 0, que é o mais baixo, ao nível 5, que é a autonomia completa [4]. Para os veículos atingirem os níveis mais altos de autonomia, a tecnologia que vem sendo implementada com sucesso é um tipo de Rede Neural Profunda, conhecida como Rede Neural Convolutacional (*Convolutional Neural Networks* - CNN). Esta técnica é formada por camadas convolucionais que tratam a imagem como uma matriz de *pixels* e aplica equações matemáticas lineares e não-lineares, gerando um modelo de variáveis e pesos otimizados. Neste trabalho, ela permite treinar um modelo inicial, a partir de um conjunto de sinais de trânsito de um país, gerar um modelo otimizado que possibilita a classificação destes sinais.

Diante dos problemas do trânsito, algumas soluções vêm sendo implementadas com sucesso nos últimos anos, tais como carros elétricos, aplicativos de trânsito, sensores e radares transmitindo informações do trânsito. Das soluções propostas, os veículos autônomos vêm se destacando como uma das mais promissoras, tendo recebido uma atenção crescente da comunidade científica, de grandes empresas de tecnologia e governantes de grandes centros urbanos.

Avanços nas técnicas de IA vem possibilitando o desenvolvimento de veículos com níveis de autonomia cada vez maiores, permitindo-lhes controlar a velocidade, estacionar, controlar a distância para outros veículos, trocar de faixa em estradas, detectar objetos na pista, entre outras ações que normalmente requerem a intervenção de um condutor humano.

Uma das principais funções desempenhadas por veículos autônomos é a detecção e identificação de objetos à sua volta. Em especial, a capacidade de identificar e interpretar placas de sinalização é essencial para que os veículos autônomos sejam capazes de evitar infrações de trânsito e acidentes. Dado isso, o problema de classificação de sinais de trânsito é abordado neste trabalho através da aplicação do aprendizado de máquina, mais especificamente, do aprendizado profundo, um ramo da IA focado em identificação de padrões a partir de exemplos.

Nesse sentido, tomando como exemplo a União Europeia ou o Mercosul, em que um veículo é autorizado a trafegar em todos os países que fazem parte daquela comunidade, tem-se um problema que é a alta variabilidade de placas e regras

de trânsito entre países. Há, então, uma preocupação com a capacidade de um veículo autônomo possuindo um modelo treinado em um determinado país de conseguir trafegar por todos os demais daquele grupo. Além disso, assumindo-se que já se tem à disposição um modelo treinado em um determinado país dessa comunidade, existe um custo associado de retrainar ou atualizar esse modelo para a configuração de placas dos outros países daquela comunidade.

O objetivo deste trabalho é verificar qual a melhor estratégia de treinamento de modelos para classificação de sinais de trânsito em diferentes países. Dado isso, vislumbra-se responder as seguintes perguntas de pesquisa neste trabalho:

- Dada a diversidade de placas de sinais de trânsito de diferentes países, que podem possuir placas iguais com a mesma semântica, placas diferentes com semânticas iguais e, ainda, placas cujo símbolo seja parecido com semânticas diferentes, um carro com um modelo de classificação de sinais de trânsito treinado em um país x de um conglomerado de países é capaz de transitar por países y ou z sem causar acidentes ou infringir a legislação de trânsito destes países?
- Supondo que dispõem-se de dados das placas de diferentes países, qual é o impacto de diferentes estratégias de treinamento de um modelo de classificação de sinais de trânsito baseado em diferentes conjuntos de dados de placas?

As contribuições deste trabalho se situam no seguinte:

1) Um mapeamento das classes de placas de trânsito de datasets públicos da Alemanha, Bélgica e Croácia, que permite que não só este como trabalhos futuros avaliem e comparem o desempenho de modelos de classificação de sinais em cada um destes países. Comparando [5] com este trabalho verificou-se que este último apresentou um número maior de classes equivalentes quando feita a comparação entre os datasets destes países e também quando foi criado o dataset único da união dos 3 datasets, verificou-se um número total de classes maior que do artigo em comparação.

2) Uma avaliação experimental extensiva para determinar a melhor estratégia de treinamento de modelos de classificação de sinais de trânsito em múltiplos países. As estratégias utilizadas foram treinamento/teste em cada dataset individualmente e treinamento/teste em um dataset único.

Este artigo está estruturado da seguinte forma: a Seção II descreve os trabalhos relacionados. A Seção III apresenta os conceitos preliminares. Por sua vez, a Seção IV explica a metodologia proposta detalhando o modelo, procedimentos de treinamento e classificação, avaliação dos resultados. A Seção V apresenta os experimentos obtidos. Por fim, a Seção VI apresenta a conclusão.

II. TRABALHOS RELACIONADOS

A Inteligência Artificial tem estado cada vez mais presente em vários setores da sociedade. Entre os avanços, destacam-se o desenvolvimento de técnicas que permitem reconhecer padrões e analisar dados. O aprendizado de máquinas, uma subárea da IA, vem desenvolvendo algoritmos que

aperfeiçoam estas técnicas. A partir destes conhecimentos, a comunidade científica vem apresentando trabalhos na área de mobilidade urbana que permitem aos veículos identificarem os sinais de trânsito, entre outras atividades, utilizando informações como alerta e orientação, de maneira que uma área que vem se beneficiando destes conhecimentos é a direção autônoma. A seguir, são apresentados trabalhos que têm relação com este artigo, seja de maneira direta ou indireta.

Em [6], foi proposto o reconhecimento de sinal de trânsito em 2 etapas: identificação de um sinal em uma imagem usando detector de região *Maximally Stable Extremal Regions*, e posterior classificação desse sinal. Os experimentos foram feitos com os seguintes datasets: *Chinese Traffic Sign Dataset* (CTSD), *German Traffic Sign Detection Benchmark* (GTSDB) e *German Traffic Sign Recognition Benchmark* (GTSRB) [7].

Em [8], é proposta uma arquitetura para classificação de sinais de trânsito chamada MCDNN (*Multi-Colum Deep Neural Network*) combinada com uma rede profunda tradicional, que permitiu o reconhecimento das imagens, mesmo aquelas com pouca iluminação ou contraste. O treinamento foi feito utilizando-se o dataset alemão GTSRB. Foi dada ênfase ao contraste das imagens e levantada a dúvida se teria os mesmos resultados para outros tipos de imagens de outros datasets.

Já [9] apresenta um modelo de classificação de sinais em que foi criado um dataset único com os dados públicos dos países Bélgica, Croácia, França, Alemanha, Holanda e Suécia. Este dataset possui 80.000 imagens divididas em 164 classes, que por sua vez distribuem-se em 4 categorias (Perigo/Aviso, Regulador, Informativo e Outras).

Em [10], é proposto um modelo de redes neurais convolucionais que usa uma arquitetura CNN com múltiplas camadas. Devido à alta variação de contraste entre as imagens, foi realizada normalização de contraste de 8 maneiras diferentes, além de ter sido utilizado o dataset público alemão GTSRB. Percebe-se, assim, que esse modelo é um tanto restrito, já que testado em apenas um conjunto de dados. [11] apresenta um método variante do *Stochastic Gradient Descent* como otimizador para treinar CNNs, apresentando uma convergência rápida e estável. O modelo foi restrito ao dataset alemão GTSRB, o que não garante bons resultados em outros datasets.

Analogamente, [12] propôs uma arquitetura de redes neurais convolucionais que permite reduzir o consumo de CPU para classificação de sinais de trânsito. O artigo obteve os melhores resultados de classificação com o dataset alemão GTSRB, mas o mesmo não generaliza bem para outros datasets. Já em [13] foi proposto três CNNs para classificação de sinais de trânsito. As redes usadas foram: uma rede padrão GoogLeNet [14], uma rede padrão VGG [15] e uma terceira que escolhe qual das duas redes anteriores vai ser usada na classificação de uma imagem a partir das características das imagens. Foi usada a base de dados de sinais de trânsito alemã GTSRB [7].

Por fim, é apresentado em [16] um algoritmo de detecção e reconhecimento de sinais de trânsito, em que consideraram o baixo desempenho em situações causadas pelo ambiente, quando usado aprendizado profundo. O artigo aponta um problema no uso do padrão HSV de cores para identificar as

placas de sinais, uma vez que são detectadas efetivamente pela forma. Assim sendo, foi proposto um modelo CNN LeNet-5 clássico, usando o kernel (Gabor) como kernel convolucional inicial, adicionando normalização em lote após a camada de agrupamento e usando algoritmo de otimização Adam [17].

III. REDES NEURAIS CONVOLUCIONAIS

Conhecidas também como CNNs (*Convolutional Neural Networks*) ou Convnets [18], possuem os princípios das redes neurais tradicionais com algumas diferenças. Nas redes tradicionais, toda entrada representada por neurônios através de matriz é multiplicada por uma matriz de pesos gerando uma saída. Nas CNNs, as convoluções são feitas em regiões específicas utilizando filtros e máscaras [1]. Assim, as CNNs têm como princípio básico a convolução que é uma operação matemática que corresponde à aplicação de um filtro nos dados de camadas anteriores em forma de matrizes. A arquitetura CNN permite extrair o mapa de características, filtrar informações, reduzir o número de parâmetros, dentre outras funcionalidades, possuindo assim, grande aplicação para imagens.

A Figura 1 mostra a arquitetura de uma rede neural convolucional composta por camadas de entradas, de convolução, de agrupamento, as completamente conectadas e as de saída. A camada inicial recebe as imagens. As camadas intermediárias extraem as características através de funções matemáticas. Inicialmente filtra linhas, curvas e bordas. Nas camadas seguintes, retiram os detalhes mais complexos. Este processo é feito percorrendo a imagem, selecionando matrizes de pixels, aplicando filtros formados por pesos que vão se atualizando através dos treinamentos. A última camada gera uma saída de tamanho igual ao número de classes treinadas que contém o percentual de acerto da imagem em cada classe.

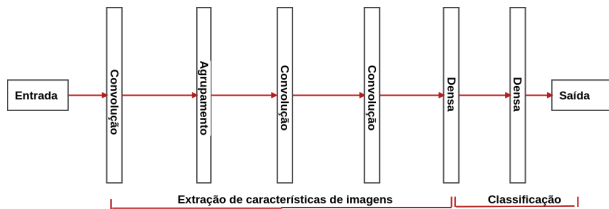


Figura 1. Arquitetura de uma Rede Neural Convolucional.

IV. MÉTODO PROPOSTO

O método proposto é composto por uma etapa de pré-processamento, em que as imagens são redimensionadas e normalizadas, e uma de classificação. Dessa forma, primeiramente é apresentado o método de pré-processamento utilizado. Na seção seguinte, é apresentado detalhadamente o modelo CNN proposto. Por fim, é descrito o método proposto para treinamento e ajuste do modelo CNN.

A. Pré-processamento

O método de pré-processamento utilizado inclui um conjunto de procedimentos para limpeza e formatação das imagens, de modo a reduzir o custo computacional na etapa

posterior e melhorar a identificação de características pelo modelo.

As transformações aplicadas e suas justificativas estão descritas a seguir. (1) A escala de cinzas possibilita criar uma uniformidade entre as imagens e identificar suas semelhanças e diferenças. (2) A normalização retira o excesso de cor, contraste e brilho. Isto permite que a imagem tenha um padrão que facilita identificar as características. (3) O contraste permite identificar melhor as imagens de fundo. (4) O redimensionamento reduz o tamanho original das imagens. Em [9] é demonstrado que as dimensões de 48×48 permite obter bons resultados em se tratando da classificação de imagens de datasets de sinais de trânsito. (5) A centralização ajuda o treinamento do modelo deslocando a imagem para o eixo central. (6) O *augmentation* é um método usado para aumentar artificialmente os dados de treinamento, de maneira a possibilitar um melhor aprendizado da rede. Para isto, são aplicadas algumas operações às imagens, tais como como translação, rotação, inclinação, ruídos, variação de brilho, saturação, *zoom* e outras. Esta técnica permite reduzir o *overfitting* e melhorar os resultados da rede sem adicionar novas características.

B. Modelo CNN

O modelo CNN proposto é composto de camadas conforme ilustradas na Figura 2. Neste modelo são aplicadas convoluções de tamanho 32, 64 e 128. A cada *maxpooling*, é reduzida a dimensão das imagens pela metade, pois, cada matriz (2, 2) calcula o valor máximo. O *dropout* quando usado não altera o tamanho das imagens utilizadas. Em seguida, é feito um *flattening* (achatamento) convertendo a saída em um vetor. A seguir, é executado uma camada densa com 512 convoluções. Por fim, executa-se uma camada densa com a função *softmax* que gera a classificação. Nas últimas linhas, observa-se o resultado específico para cada país.

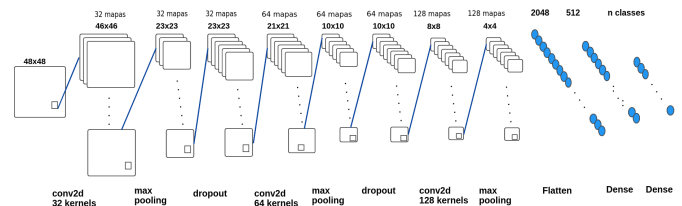


Figura 2. Modelo CNN, baseado em [19].

O modelo adota *categorical_crossentropy* como função de erro, pelo fato de o problema ser de classificação. Por sua vez, o otimizador utilizado é o *Stochastic Gradient Descent*.

C. Treinamento

No treinamento, os seguintes valores foram usados: *epoch* = 30 e *validation* de 20% e 30%, visto que se observou que estes valores obtiveram as melhores acurácias. Os seguintes hiperparâmetros foram combinados e variados, a fim de se descobrir a combinação que obtém o melhor desempenho: *batch_size*, *augmentation* e *dropout*. Por fim, para responder as perguntas

de pesquisa estabelecidas, várias estratégias de treinamento foram testadas, conforme será explicado na Seção V.

Para cada experimento, após a etapa de treinamento, é realizada a etapa de teste. O modelo CNN obtido é então aplicado aos dados de teste. Dessa forma, são obtidas as métricas acurácia, *precision*, *recall* e *f1 score*, de modo a constatar o desempenho do modelo.

V. EXPERIMENTOS COMPUTACIONAIS

O objetivo dos experimentos foi verificar como se pode avaliar a classificação de sinais de trânsito por veículos autônomos, fazendo uso de algoritmos de aprendizado de máquina e se deslocando por diferentes países.

A. Ambiente Computacional

O ambiente computacional usado na implementação foi um Notebook Gamer Dell com 16 GB de memória, processador i7-8750H, placa gráfica Nvidia GeForce GTX 1060 com 6GB e o sistema operacional Ubuntu 16 Linux. Na parte de software, a implementação dos programas foi feita na linguagem Python 3.7 com interface CUDA para GPU,

B. Datasets

Os datasets de sinais de trânsito utilizados nos experimentos foram: Alemão, Belga e Croata. O conjunto de sinais de trânsito alemão, conhecido como GTSRB [7] é composto por um conjunto de dados de treinamento com 31.367 imagens, uma base de teste com 12.630 imagens, sendo que cada base de dados contém 43 classes distintas.

Já o dataset BTSC [20] reúne os sinais de trânsito da Bélgica, sendo composto por uma conjunto de treinamento contendo 4.591 imagens, um de teste contendo 2.534 imagens e cada base de dados contém 62 classes.

Por sua vez, o conjunto de imagens representativas de sinais de trânsito da Croácia rMASTIF [21] foi criado pelo MASTIF (*Mapping and Assessing the State of Traffic InFrastructure*), sendo composto por um conjunto de treinamento contendo 4.044 imagens e uma de teste contendo 1.784 imagens. Cada base de dados contém 31 classes.

C. Experimento 1 - Treinamento Individual

Esta primeira estratégia de treinamento foi usada para responder à questão da pesquisa que procura saber qual é o impacto de diferentes estratégias de treinamento do modelo proposto em cima de diferentes conjuntos de dados de placas. Assim, este treinamento é constituído de experimentos feitos para os datasets de treino e teste de cada país individualmente. Dessa forma, primeiramente é feita a leitura do dataset de treino passando por pré-processamento para deixar os dados normalizados e redimensionados. Em seguida, o modelo CNN faz o treinamento gerando um arquivo contendo o valor dos parâmetros obtidos e que é usado na etapa de teste para obter as métricas de avaliação.

Nas subseções seguintes, são apresentados os resultados do treinamento individual dos datasets utilizados.

1) *Dataset Alemão*: Foi feito o treinamento do modelo proposto sobre o dataset alemão (GTSRB) [7].

A Tabela I mostra os hiperparâmetros usados no treinamento com dataset de treino e a acurácia obtida quando aplicado este modelo no dataset de teste. A melhor acurácia foi de 98.89% obtida quando usou *batch size* de 24, *dropout* de 0.2 e *augmentation* de 20%. Neste caso, *precision* foi de 98.49%, *recall* foi de 98.74% e *f1 score* foi de 98.58%.

Tabela I
ACURÁCIA OBTIDA PARA DIFERENTES CONFIGURAÇÕES NO TREINAMENTO DO MODELO PARA O DATASET ALEMÃO.

<i>Batch size</i>	16	24	32	48
sem drop	93.94	95.70	95.24	94.95
augm 20 + val 20	97.77	98.17	98.15	97.78
augm 20 + val 30	97.96	98.13	97.74	97.71
drop 0.2	94.53	97.15	97.25	97.15
drop 0.2 + augm 20	97.67	98.89	98.67	98.36

2) *Dataset Belga*: Neste treinamento foi utilizado o modelo CNN proposto com o dataset belga BTSC [20].

A Tabela II mostra os hiperparâmetros usados no treinamento com dataset de treino e a acurácia obtida quando aplicado este modelo no dataset de teste. A melhor acurácia foi de 97.77% obtida quando usou *batch size* de 8, *dropout* de 0.2 e *augmentation* de 20%. Por sua vez, *precision* foi de 87.80%, *recall* foi de 89.06% e *f1 score* foi de 98.58%.

Tabela II
ACURÁCIA OBTIDA PARA DIFERENTES CONFIGURAÇÕES NO TREINAMENTO DO MODELO PARA O DATASET BELGA.

<i>Batch size</i>	8	16	24	32
sem drop	93.96	96.66	96.38	96.70
augm 20 + val 20	93.80	96.42	95.59	94.04
augm 20 + val 30	97.61	97.46	96.38	96.54
drop 0.2	95.95	96.62	96.34	96.07
drop 0.2 + augm 20%	97.77	97.02	96.19	95.67

3) *Dataset Croata*: Para este treinamento foi usado o modelo CNN proposto com o dataset Croata rMASTIF [21].

A Tabela III mostra os hiperparâmetros usados no treinamento com dataset de treino e a acurácia obtida quando aplicado este modelo no dataset de teste. A melhor acurácia foi de 97.36% obtida quando usou *batch size* de 16, *augmentation* de 20% e *validation* de 30%. Por seu turno, *precision* foi de 96.95%, *recall* de 96.08% e *f1 score* de 96.36%

Tabela III
ACURÁCIA OBTIDA PARA DIFERENTES CONFIGURAÇÕES NO TREINAMENTO DO MODELO PARA O DATASET CROATA.

<i>Batch size</i>	8	16	24	32
sem drop	91.42	93.94	94.73	94.22
augm 20 + val 20	96.13	95.90	97.02	95.17
augm 20 + val 30	94.61	97.36	96.58	91.81
drop 0.2	95.57	96.74	95.79	95.29
drop 0.2 + augm 20	96.58	96.80	94.95	94.84

4) *Resumo*: A Tabela IV mostra o resumo das configurações e dos valores das métricas de avaliação

para o treinamento individual dos datasets Alemão, Belga e Croata. Em todos os casos, o *data augmentation* esteve presente. A maior acurácia foi obtida para o dataset Alemão, em que as imagens possuem mais representatividade e, portanto, possibilitam um melhor aprendizado do modelo. Verificou-se que o modelo CNN proposto se saiu bem nesta estratégia de treinamento individual de cada um dos 3 datasets, obtendo valores de acurácia acima de 97% para as melhores configurações. Os menores valores das demais métricas foram obtidos para o dataset Belga em que o modelo, apesar de atingir 97.77% de acurácia, obteve 87.80% e 89.06% para *precision* e *recall*, respectivamente.

Tabela IV

RESUMO DOS VALORES OBTIDOS DE CADA MÉTRICA PARA A MELHOR CONFIGURAÇÃO DE HIPERPARÂMETROS, PARA OS DATASETS TESTADOS.

Dataset	Acur	Prec	Rec	BS	Hiperparâmetros
Alemão	98.89	98.49	98.74	24	Drop 0.2 + Augm de 20%
Belga	97.77	87.80	89.06	8	Augm de 20% + val de 20%
Croata	97.36	96.95	96.08	16	Augm de 20% + val de 30%

BS = Batch Size

D. Experimento 2 - Treinamento do Dataset Único

Esta estratégia corresponde à segunda estratégia usada para responder à questão da pesquisa em que se busca saber o impacto de diferentes estratégias de treinamento do modelo proposto em cima de diferentes conjuntos de dados de placas. Para realizar esta estratégia de treinamento, primeiramente foi feito um mapeamento entre as classes equivalentes, isto é, de mesma semântica, dos datasets Alemão, Belga e Croata.

Após isso, foi criado um dataset único de treino com a união das imagens dessas classes equivalentes dos três datasets. A união desses datasets foi feita da seguinte forma: inicialmente criou-se um conjunto de dados a partir da base alemã. Neste novo dataset, foram importadas as imagens das classes Belga e Croata que tinham correspondência com este dataset. Para as imagens que não tinham correspondência, foi criada uma classe especial. Por fim, o dataset final ficou com 88 classes.

A estratégia do experimento 2, em que o treinamento é feito no dataset único, corresponde à união obtida das bases de treinamento dos três datasets, e o teste é feito de quatro maneiras: (i) na base de teste do Alemão; (ii) na base de teste do Belga; (iii) na base de teste do Croata; e (iv) na união das bases de teste dos três datasets.

1) *Teste das bases Alemão, Belga, Croata*: Neste experimento, foi usada a base única, obtida através da união dos três datasets, para as etapas de treino e teste, fazendo a devida separação. Os resultados estão exibidos na Tabela V. A melhor configuração foi obtida usando-se *batch size* igual a 24, *augmentation* de 20% e *validation* de 20%, chegando-se à acurácia de 98.37%. O tempo de treinamento dessa configuração foi de 666s. Além disso, *precision* foi de 94.61%, *recall* de 96.82% e *f1 score* de 95.20%.

Tabela V

ACURÁCIA OBTIDA PELO MODELO TREINADO E TESTADO NO DATASET DA UNIÃO.

Batch size	16	24	32	48
drop	93.10	96.58	96.97	96.01
augm 20 + val 20	95.60	98.37	98.22	98.26
augm 20 + val 30	97.07	97.91	97.56	97.67
drop 0,2	91.54	96.25	96.25	96.16
drop 0,2 + augm 20	95.05	98.13	98.20	98.32

2) *Teste da base Alemã*: Neste experimento, foi usada a base única para a etapa de treino e a base alemã para a etapa de teste. Os resultados são mostrados na Tabela VI. A melhor configuração foi usando *batch size* igual a 48, *dropout* de 0.2 e *augmentation* de 20%, obtendo-se o valor de 98.59% para a acurácia. O tempo de treinamento dessa configuração foi de 621s. Ademais, *precision* foi de 80.98%, *recall* de 81.26% e *f1 score* de 81.26%.

Tabela VI

ACURÁCIA OBTIDA PELO MODELO TREINADO NO DATASET DA UNIÃO E TESTADO NO DATASET ALEMÃO.

Batch size	16	24	32	48
drop	93.27	96.19	95.60	95.35
augm 20 + val 20	97.62	98.40	98.24	98.13
augm 20 + val 30	97.74	98.25	97.80	97.68
drop 0,2	94.18	97.24	96.96	96.50
drop 0,2 + augm 20	96.56	98.55	98.56	98.59

3) *Teste da base Belga*: Analogamente, neste experimento, foi utilizada a base única para realizar o treinamento e a base belga, para o teste. Os resultados obtidos estão na Tabela VII. A melhor configuração foi obtida usando-se *batch size* igual a 24, *dropout* de 0.2 e *augmentation* de 20%, chegando-se à acurácia de 97.18%. O tempo de treinamento dessa configuração foi de 584s. Por sua vez, *precision* foi de 85.11%, *recall* de 84.65% e *f1 score* de 84.45%.

Tabela VII

ACURÁCIA OBTIDA PELO MODELO TREINADO NO DATASET DA UNIÃO E TESTADO NO DATASET BELGA.

Batch size	16	24	32	48
drop	90.59	92.38	91.86	93.21
augm 20 + val 20	96.19	97.14	96.94	96.54
augm 20 + val 30	96.62	97.06	96.50	96.74
drop 0,2	91.34	93.80	94.20	94.00
drop 0,2 + augm 20	91.94	97.18	96.74	97.02

4) *Teste da base Croata*: Por sua vez, neste experimento, foi feita a mesma coisa, mas trocando-se a base de teste para o dataset Croata. Os resultados obtidos estão expostos na Tabela VIII. A melhor configuração utilizou *batch size* igual a 32, *augmentation* de 20% e *validation* de 20%, obtendo acurácia de 98.82% e tomando 486s como tempo de treinamento. Além disso, *precision* foi de 92.60%, *recall* de 92.24% e *f1 score* de 92.36%.

Tabela VIII
ACURÁCIA OBTIDA PELO MODELO TREINADO NO DATASET DA UNIÃO E TESTADO NO DATASET CROATA.

Batch size	16	24	32	48
drop	91.98	95.01	95.06	95.12
augm 20 + val 20	96.74	98.43	98.82	98.31
augm 20 + val 30	96.18	98.65	98.31	98.43
drop 0,2	91.36	96.63	96.91	96.13
drop 0,2 + augm 20	93.77	98.15	98.76	98.76

5) *Resumo*: A Tabela IX mostra o resumo do experimento em que o modelo é treinado no dataset referente à união dos datasets Alemão, Belga e Croata. O teste foi feito no dataset único e também individualmente nos datasets Alemão, Belga e Croata, respeitando as devidas separações. Esta tabela mostrou que se obtém melhor acurácia quando se tem uma base de treino rica e diversa em imagens com boa qualidade e variações, como foi apontado por [1], visto que isto permite cobrir uma maior quantidade de características e permite ao modelo aprender a extraí-las. Fica claro que a estratégia de se adotar um dataset único contendo imagens de sinais de trânsito dos países de uma determinada comunidade é a melhor opção, respondendo à pergunta de pesquisa levantada.

Tabela IX
RESUMO DOS EXPERIMENTOS COM O DATASET REFERENTE À UNIÃO.

Dataset Teste	Acurácia	Precision	Recall	BS	Hiperparâmetros
AlBeCr	98.37	94.61	96.82	24	drop 0,2+aug20%
Alemão	98.59	80.98	81.26	32	drop 0,2+aug20%
Belga	97.18	85.11	84.65	32	drop 0,2+aug20%
Croata	98.82	92.60	92.24	48	drop 0,2+aug20%

AlBeCr=AlemãoBelgaCroata BS=Batch Size

VI. CONCLUSÃO

Como esperado, modelos de aprendizado profundo tiram maior proveito de uma base de dados mais rica e diversa em imagens, conforme visto no desempenho do modelo para o dataset alemão que apresentou 98,89% de acurácia, superior ao belga com 97,77% e croata com 97,36%, ainda que a diferença seja relativamente pequena. Isso se deve ao fato de o dataset Alemão possuir um maior número de imagens representativas por classes tanto na base de treino quanto na de teste, que permitiu ao modelo extrair mais características e ter um melhor aprendizado. Por sua vez, quando se criou um dataset único a partir da união dos datasets, treinou-se o modelo nessa união e testou-se em cada um dos três datasets. Foi obtida uma alta acurácia para os três datasets: Alemão - 98,40%, Belga - 97,30%, Croata - 98,76%. Dessa forma, conclui-se que o melhor modelo foi aquele treinado com dataset referente à união, visto que os resultados foram tão bons quanto os obtidos pela estratégia de treinamentos individuais, com a vantagem de possibilitar o deslocamento entre os países elencados. Ressalte-se que não existe ainda um modelo que consiga contemplar todos os sinais de trânsito de diversos países e obter valores de acurácia, *precision* e *recall* próximos de 100%.

REFERÊNCIAS

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- [3] S. M. Weiss and C. A. Kulikowski, *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Morgan Kaufmann Publishers Inc., 1991.
- [4] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *arXiv preprint arXiv:1905.12762*, 2019.
- [5] F. Jurišić, I. Filković, and Z. Kalafatić, "Multiple-dataset traffic sign classification with onecnn," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2015, pp. 614–618.
- [6] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2022–2031, 2015.
- [7] "Dataset alemão de sinais de trânsito," <http://benchmark.ini.rub.de/?section=gtsrbsubsection=dataset>, accessed: 29/03/2020.
- [8] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural networks*, vol. 32, pp. 333–338, 2012.
- [9] C. G. Serna and Y. Ruichek, "Classification of traffic signs: The european dataset," *IEEE Access*, vol. 6, pp. 78 136–78 148, 2018.
- [10] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1918–1921.
- [11] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1991–2000, 2014.
- [12] H. H. Aghdam, E. J. Heravi, and D. Puig, "A practical and highly optimized convolutional neural network for classifying traffic signs in real-time," *International Journal of Computer Vision*, vol. 122, no. 2, pp. 246–269, 2017.
- [13] S. Saha, M. S. Islam, M. A. B. Khaled, and S. Tairin, "An efficient traffic sign recognition approach using a novel deep neural network selection architecture," in *Emerging Technologies in Data Mining and Information Security*. Springer, 2019, pp. 849–862.
- [14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [16] J. Cao, C. Song, S. Peng, F. Xiao, and S. Song, "Improved traffic sign detection and recognition algorithm for intelligent vehicles," *Sensors*, vol. 19, no. 18, p. 4021, 2019.
- [17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [20] "Dataset belga de sinais de trânsito," <https://btsd.ethz.ch/shareddata/>, accessed: 29/03/2020.
- [21] "Dataset croata de sinais de trânsito," <http://www.zemris.fer.hr/kalfa/Datasets/rMASTIF/>, accessed: 29/03/2020.