MAPEAMENTO DE OBJETOS NO FUNCIONAMENTO DE SISTEMAS AUTÔNOMOS

Bruno Seki Schenberg (IC) e Leandro Carlos Fernandes (Orientador)

Apoio: PIBIC Mackenzie

RESUMO

O presente estudo tem por objetivo analisar o estado atual de detecção e classificação de sinalização vertical de trânsito por meio de inteligência artificial, construir um modelo de inteligência artificial para executar esta mesma tarefa, e também avaliar a sua performance. Para a construção do projeto foram estudados diferentes meios de detecção de objetos de interesse em imagens por meio de inteligência artificial. Foram utilizados dois datasets específicos de placas de trânsito para a construção do projeto. O primeiro dataset é um específico para a tarefa de classificação, chamado de GTSRB, e o segundo dataset é um específico para a tarefa de detecção, chamado de GTSDB. Ambos foram produzidos pelo instituto de neuroinformática da universidade de Ruhr de Bochum. O projeto foi construído na linguagem python com o uso das bibliotecas keras e YOLO, utilizando também a plataforma Roboflow de apoio. A resposta final gerada pelo programa é feita pelo trabalho em conjunto de dois modelos distintos de inteligência artificial, onde a imagem é fornecida para o programa, onde primeiro são feitas as detecções. Após feitas essas, elas são fornecidas para uma segunda rede neural, que tem o trabalho de classificar o objeto detectado como uma das placas de trânsito alemãs.

Palavras-chave: Inteligência Artificial. Veículo. Autônomo.

ABSTRACT

The present study aims to analyze the current state of detection and classification of traffic signs through the means of artificial intelligence, build an a.i. model to perform this task, and also evaluate its performance. For the construction of the project different methods of object detection in images with the use of a.i. were studied. Two traffic signs specific datasets were used in this project. The first dataset, GTSRB is one specific for classification, and the second dataset, GTSDB is one specific for the task of detection. Both were produced by the neuroinformatics institute from the Ruhr University of Bochum. The project was constructed in python with the use of the keras and YOLO libraries, with the Roboflow platform for support. The Final answer generated by the program is made by the combined work of two distinct a.i. models, where the image is supplied to the program, and first the detections are made on the image. Those detections are then supplied to a second neural network that has the job of classifying the objects detected as one of the German traffic signs.

Keywords: Artificial Intelligence. Autonomous. Vehicle.

1. INTRODUÇÃO

Os ambientes urbanos, principalmente no que tange o tráfego de veículos e de pedestres, caracterizam-se por serem complexos, dinâmicos e diversificados quanto a seus participantes. As leis de trânsito, além de promover um convívio seguro e funcional, buscam regulamentar, organizar e dar fluidez para os veículos, pedestres e demais integrantes do trânsito nas cidades.

Nesse contexto, a sinalização é um elemento essencial para que haja ordem, organização e segurança entre os envolvidos. Através dela temos a determinação de limites de velocidade, preferência, direção permitida, dentre outros, para os veículos que trafegam em meios as vias.

De acordo com dados da Organização Mundial de Saúde, 1,35 milhão de mortes ocorreram nas estradas em 2018, a maioria das vítimas pedestres, ciclistas e motociclistas (OMS, 2021). Esta é a maior causa de morte de pessoas entre 5 e 29 anos no mundo. Dentre as principais causas de acidentes de trânsito, a Organização Mundial de Saúde destaca dirigir em alta velocidade, dirigir sob influência de álcool e outras substâncias psicoativas, dirigir distraído, entre outras (OMS 2022). Por causa disso, o uso de tecnologias para auxiliar o motorista a ter as informações corretas na estrada poderia contribuir para evitar acidentes.

No relatório técnico publicado em 2010 pelo Instituto de Padrões de Telecomunicações Europeu (ETSI, 2010), uma das principais preocupações do motorista é o conhecimento dos seus arredores próximos, e com isso estão inclusos a atenção para a sinalização vertical, características da via, condições de tráfego, como presença de outros veículos, pedestres e obstáculos, e condições meteorológicas. Desta forma, uma aplicação de software, que auxilia o motorista e o avisa sobre a sinalização vertical detectada à frente do veículo não só pode prevenir acidentes causados por falta de atenção, como também é um dos passos mais importantes para construção de um carro cem por cento autônomo.

A importância do projeto é uma tentativa de reduzir acidentes de trânsito, que atualmente representam um perigo significativo à vida de muitas pessoas.

Assim, a questão tecnológica a ser respondida é: há uma forma de interpretar a sinalização vertical, tais como placas de trânsito, através do processamento de imagens, com o objetivo de fornecer ao motorista informações presentes no cenário? Isso seria feito com a utilização de inteligência artificial, que operaria sobre as imagens da rua, segmentando a imagem e detectando as placas, e posteriormente

as classificando, e assim fornecendo uma resposta baseada nas detecções feitas na imagem.

Portanto, o tema desenvolvido neste trabalho foi o estudo da utilização de inteligência artificial, para reconhecimento de sinalização vertical de trânsito, em específico placas, com o uso de redes neurais convolucionais, para posteriormente conseguir implementar este sistema em um veículo e assim gerar uma resposta para auxiliar o motorista em receber as informações que lhe são relevantes.

A pesquisa pode ser dividida em 3 etapas, a primeira sendo estudo sobre o funcionamento das redes neurais convolucionais, e o estudo da tecnologia e resultados de outras pesquisas na mesma área, a segunda etapa a de implementação, e a terceira etapa a análise dos resultados obtidos e comparação com resultados de outras técnicas aplicadas com o mesmo intuito, de detectar e classificar a sinalização vertical de trânsito.

Mais especificamente, a implementação do software foi dividida em 3 partes, sendo a primeira parte o estudo e desenvolvimento de uma rede neural convolucional capaz de classificar as placas de trânsito, treinada e testada no dataset GTSRB (*German Traffic Sign Recognition Benchmark*), a segunda parte o estudo, treinamento e uso do modelo de rede neural YOLO, treinado e testado no dataset GTSDB (*German Traffic Sign Detection Benchmark*) para detectar as imagens, e a terceira parte a integração das duas redes trabalhando em conjunto, para obter os resultados desejados.

2. REFERENCIAL TEÓRICO

Esta pesquisa se encontra dentro de duas áreas da computação: a área de processamento de imagens, que aborda como computadores conseguem processar e manipular imagens digitais; e a área de inteligência artificial, que estuda e desenvolve técnicas para o reconhecimento e identificação de padrões, através de algoritmos de machine learning. São essas técnicas de inteligência artificial que permitem “treinar” as máquinas a resolverem problemas complexos, tal como o problema abordado nesta pesquisa, o de reconhecimento de objetos em imagens.

A visão computacional existe já desde a década de 60, quando universidades estavam começando os estudos com inteligência artificial, e almejavam conseguir com que o computador pudesse enxergar como os humanos. No entanto, a área conseguiu evoluir tremendamente nas seis décadas decorridas desde então, com muito do que se esperava na época sendo atingido nos últimos anos.

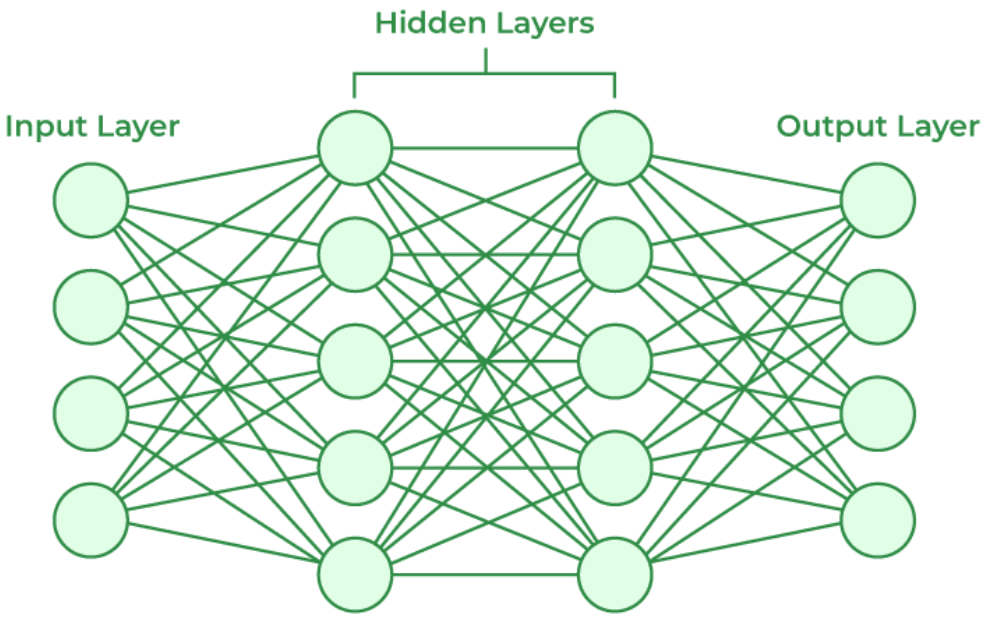
A maneira que as imagens são processadas pelo computador começa pelo hardware, onde se encontram os mais variados sensores para captar imagens de maneira digital. A mais comum é a câmera com sensor óptico, que como output cria 3 matrizes, de valores numéricos que representam os pixels, nos seus valores de vermelho, verde e azul. O tamanho da matriz representa a resolução da imagem, e assim, é possível manipular a imagem para um tamanho menor, de forma que ela perca qualidade dos detalhes, mas seja também mais rápida para ser processada pelo computador, já que será representada por uma matriz menor. É possível também reduzir a imagem à uma escala de cinza, para que se trabalhe com apenas uma matriz em vez de 3, também com o objetivo de tornar o seu processamento mais rápido. Isto foi explorado na rede Deepthin (HAQUE, 2021)

A forma que as imagens são processadas pelas redes neurais difere de um método para outro. Os sistemas de detecção clássicos reutilizam sistemas de classificação com o propósito de detectar. A maneira que isso era feito era alimentando a rede neural classificadora com pedaços diferentes em posições diferentes da imagem, em busca de objetos de interesse, um dos problemas disso é que era muito custoso em termos de tempo e processamento. Um método mais recente é com o uso de R-CNNs, ou *Region based Convolutional Neural Network*, uma rede neural mais especializada em detecções. A forma que ela opera é gerando *“bounding boxes”* em localizações potenciais, e então utilizando um classificador nestas caixas, e depois disso um processamento final para refinar as caixas e eliminar detecções duplicadas.

A maneira que o modelo YOLO realiza detecções é “tratando detecção como um problema de regressão, diretamente dos pixels das imagens para as coordenadas das caixas e probabilidades de classe.” (REDMON, 2016) A forma que isto é feita é dividindo a imagem em uma grade e predizendo as caixas para cada classe baseado em sua probabilidade, e posteriormente utilizando a técnica de NMS, acrônimo de *non maximum suppression*, onde caixas duplicadas que estão sobrepostas são eliminadas, mantendo apenas uma instância de caixa por objeto. (TERVEN, 2023)

Uma questão que permanece é: como que o programa pode aprender a predizer objetos e localizá-los na imagem? Isto é resultado do uso de algoritmos de machine learning, onde é possível treinar a máquina a resolver um problema. Na Figura 1 se encontra um modelo generalizado de uma rede neural:

Figura 1: Modelo geral de uma rede neural



Fonte: Geeksforgeeks, disponível em: https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/

O funcionamento de uma rede neural é caracterizado por 3 aspectos principais: a função de ativação, que é uma função matemática para decidir se o neurônio será ativado ou não, a interconexão dos neurônios, e o ajuste de pesos. O resultado da função de ativação depende dos valores que um neurônio recebe dos neurônios que o precedem, presentes na camada anterior, e dos pesos atribuídos a esses valores. Dessa forma é calculado o seu resultado, que é propagado à frente. Quando a rede neural está sendo treinada, o *feedback* referente a acertos ou erros ajusta os pesos da rede, e esta correção promove o seu aprendizado.

No caso de redes de classificação, a camada de *output* é composta por vários neurônios, onde cada um representa uma das classes, e assim o neurônio que possuir o maior valor nesta camada seria a classe que está sendo predita pela rede neural. A camada inicial, no caso de redes sendo utilizadas em classificação de imagem, pega os valores numéricos de cada pixel da imagem.

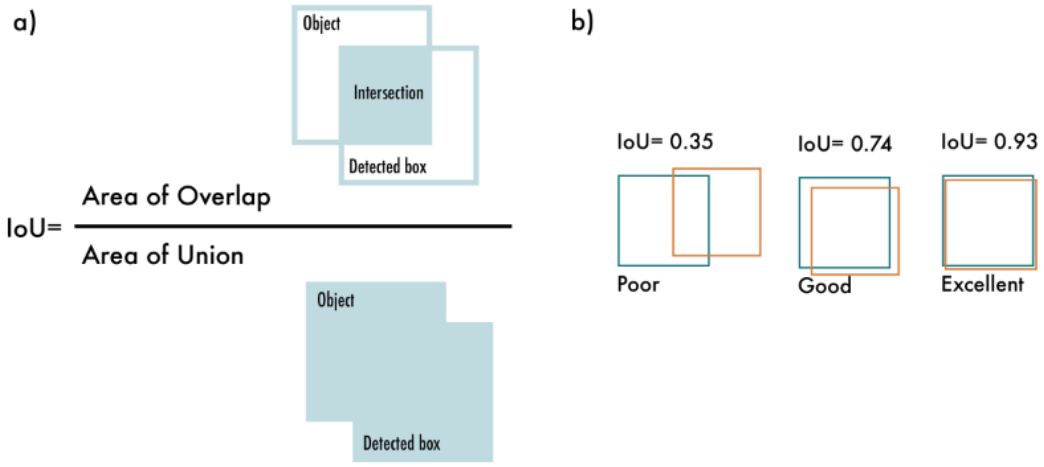
É muito importante que se possa avaliar a qualidade da performance de um modelo, e para isso se utilizam diferentes métricas e indicadores. No caso de muitos modelos classificadores, e também do YOLO, a métrica de avaliação utilizada é a mAP, ou *mean average precision*, que é composta por 3 métricas. A primeira é uma composição dos indicadores *precision* e *recall, a segunda* métrica é a capacidade de lidar com múltiplas classes de objetos, e a terceira métrica é o IoU (*Intersection Over Union*).

O indicador *precision* descreve os verdadeiros positivos em meio a todas as predições positivas. Já o *recall* representa os verdadeiros positivos em meio a todos os casos verdadeiros. Ambos são comumente empregados na área *machine learning*, servindo como referência para avaliação de desempenho a diversos algoritmos.

Por sua vez, a capacidade de lidar com múltiplas classes de objetos é também importante para a avaliação de um modelo. Esta é descrita pelo cálculo da média entre os resultados de *precision* e *recall* do modelo para cada classe de objeto.

Já o IoU, ou *intersection over union*, é a métrica utilizada para avaliar o quão bem o modelo consegue localizar os objetos. É calculado pela divisão da área de intersecção entre a caixa predita e a caixa real do objeto, em relação a área de união entre a caixa predita e a caixa real do objeto (vide Figura 2a). Quanto maior o valor, melhor é a localização do elemento de interesse, conforme ilustrado na Figura 2b, em que se apresentam exemplos de IoUs ruins, bons e excelentes.

Figura 2: Cálculo de IoU e exemplos de IoUs diferentes



Fonte: (TERVEN, 2023).

3. METODOLOGIA

O tipo de pesquisa adotada foi a de pesquisa descritiva, buscando as técnicas mais atuais na utilização de inteligência artificial como ferramenta para detecção e classificação de placas de trânsito, encontrando os datasets (conjuntos de dados específicos) mais apropriados, e por final, comparando os resultados obtidos com outras técnicas utilizadas no mesmo meio.

É importante notar, que para uma comparação justa da precisão e eficiência de uma técnica empregada para detectar ou classificar um conjunto de dados,

ambas as técnicas devem utilizar o mesmo conjunto de dados, para que se tenha uma comparação sem vieses onde um conjunto de dados pode apresentar imagens mais fáceis de se classificar, e o outro um conjunto de dados que seria considerado mais difícil, e por causa disso existe a importância dos datasets de benchmarking.

O primeiro de dois datasets utilizados nesta pesquisa é o GTSRB, um dos datasets mais populares no estudo de classificação de imagem por meio de redes neurais convolucionais. No website kaggle, existem atualmente mais de 250 notebooks de código, onde as pessoas almejam atingir os melhores percentuais de precisão, métrica importante para comparar a qualidade e eficiência da rede neural sendo aplicada.

O dataset conta com mais de 50 mil imagens no total, distribuídas em 42 classes, onde cada classe é um tipo de placa de trânsito. Neste dataset todas as imagens o conteúdo integral é uma placa de trânsito alemã. Isso difere do segundo dataset utilizado, onde as imagens são de uma rua ou estrada, onde as placas não estão ocupando o enquadramento total da imagem.

Este dataset foi utilizado para comparação da precisão de classificação da rede neural, onde a rede já é alimentada com o enquadramento adequado da imagem da placa, para fornecer os resultados, não possuindo papel de encontrar as placas de trânsito dentro de um contexto real, onde seria empregada uma câmera montada no teto do veículo, visualizando a estrada à sua frente como um todo.

O segundo dataset utilizado foi o GTSDB, que possui imagens de estrada com placas de trânsito, e também anotações da localização das placas de trânsito em cada imagem, e sua classificação. Ele conta com 900 imagens, distribuídas entre 43 classes. A classe adicional que este dataset possui, que não está presente no GTSRB é a classe de background. Isto significa que na imagem não há nenhuma placa de trânsito, e é importante que a rede neural não aponte falsos positivos, e por tanto imagens sem placas são bastante importantes para a avaliação de performance da rede neural.

Este segundo dataset, necessitava de algumas especificações para ser utilizado no trabalho. A primeira, é de que contivesse imagens de placas de trânsito do mesmo padrão que o primeiro, no caso as placas de trânsito alemãs. A segunda especificação necessária é de que fosse um dataset de detecção, com imagens onde a placa de trânsito tem que ser encontrada pela rede neural. A terceira e última especificação é a de que o dataset precisa conter as anotações, de classe e localização de cada objeto de interesse na imagem, neste caso, as placas de

trânsito, desta forma é possível treinar a inteligência artificial e verificar se a inteligência artificial consegue apontar em que lugar está a placa e dizer a sua classe. Isto é necessário para que fosse simulado o funcionamento do programa em um cenário real, onde a câmera vai apenas fornecer uma imagem da rua, e a inteligência artificial será responsável por localizar a placa. Este dataset e o primeiro, foram construídos e fornecidos ao público pelo INI (Institut für Neuroinformatik), da universidade Ruhr de Bochum.

Este estudo utilizou de validação cruzada para avaliar os modelos produzidos. A validação cruzada permite avaliar a capacidade de generalização de um modelo. Ela é feita particionando o conjunto de dados, para que uma parte seja utilizada para treinamento e a outra parte, de dados não ainda vistos pela rede sejam utilizados para validação do modelo.

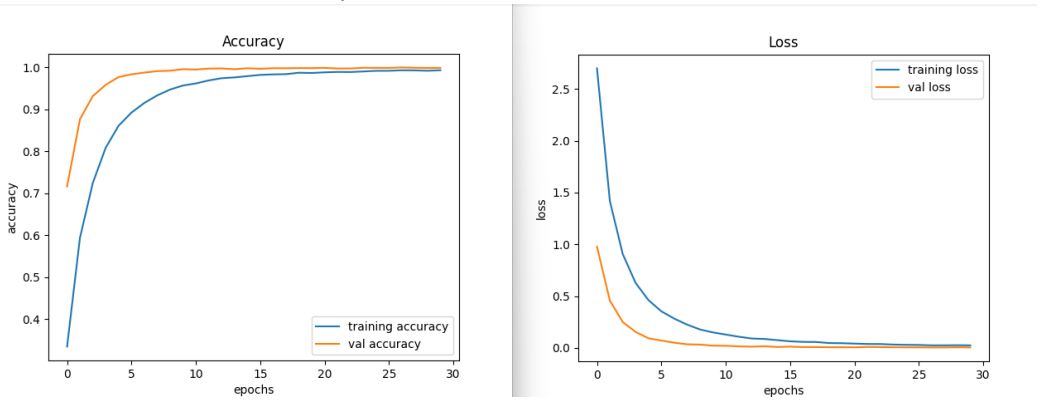
4. RESULTADO E DISCUSSÃO

O estudo e desenvolvimento da primeira rede neural do projeto, feita utilizando a biblioteca do keras, escrita em python, serviu não só de aprendizado, mas também como complemento para o resultado da segunda rede neural do projeto, utilizando a biblioteca do YOLO, que por si só foi incapaz de trazer resultados suficientes, do ponto de vista de mAP *(mean average precision)*. Isso não é uma falha do modelo YOLO, na tarefa de detecção, mas apenas uma falha na quantidade e qualidade de dados do dataset de detecção utilizado, que precisou de ajustes ao longo do desenvolvimento para trazer resultados melhores, e no final, o emprego da primeira rede desenvolvida para auxiliar na geração da resposta final.

A primeira rede desenvolvida evoluiu sem muitos obstáculos, atingindo métricas de precisão e recall na média das atingidas por outros notebooks no kaggle utilizando este mesmo dataset. O único ajuste feito nas imagens foi o redimensionamento para o tamanho de 32x32 pixels, que são processadas em cores, utilizando 3 matrizes de 32x32 como entrada da rede.

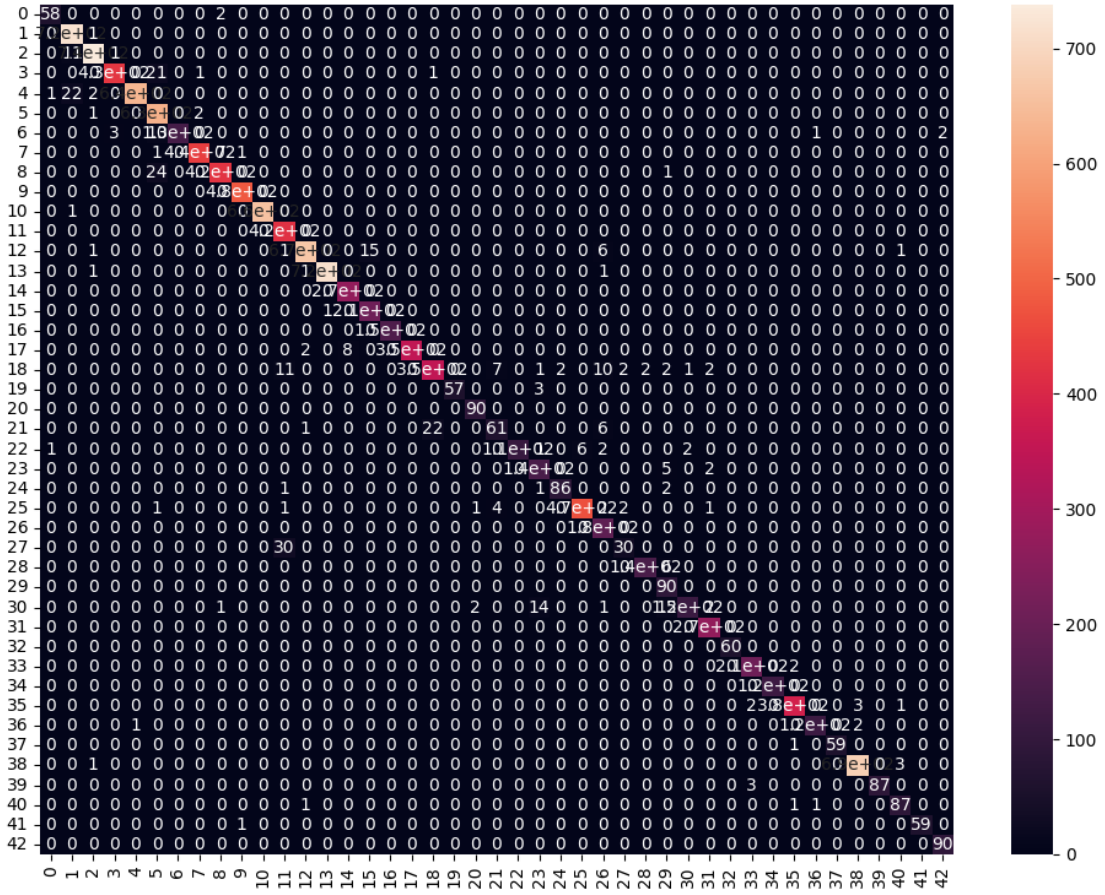
Na figura 3 são apresentados os gráficos de acurácia e perda da rede classificadora ao longo de suas 30 épocas de treinamento, e na figura 4 a matriz de confusão da rede classificadora.

Figura 3: Gráficos de acurácia e perda da rede classificadora



Fonte: elaborado pelo autor.

Figura 4: Matriz de confusão da rede classificadora



Fonte: elaborado pelo autor.

Como apresentado acima, esta rede possui um bom desempenho de classificação, atingindo a média de 0.97 de f1 score.

Como mencionado anteriormente, a função desta rede neural é de classificar as placas detectadas pela outra rede neural, feita no modelo YOLO, e não apresenta tamanha utilidade prática sozinha.

Já a segunda rede neural que foi desenvolvida neste trabalho, tem a tarefa de produzir as detecções. O primeiro passo era definir como trabalhar com detecção de

regiões de interesse em imagens, e assim o modelo YOLO aparentou ser o mais apropriado para este propósito. O grande diferencial do modelo YOLO, é a sua rapidez de tempo de resposta, para produzir as inferências. No contexto dos carros autônomos, é crítico que se detecte as placas com acurácia e velocidade, pois caso o grau de latência for elevado, é possível que já tenha sido tarde demais para uma detecção avisar o motorista.

A velocidade de resposta elevada da arquitetura YOLO, é devido ao seu funcionamento, que também é mencionado em seu nome, que é uma sigla para "*You Only Look Once*", onde a imagem é segmentada e processada toda de uma vez só, trazendo ganhos de velocidade de performance.

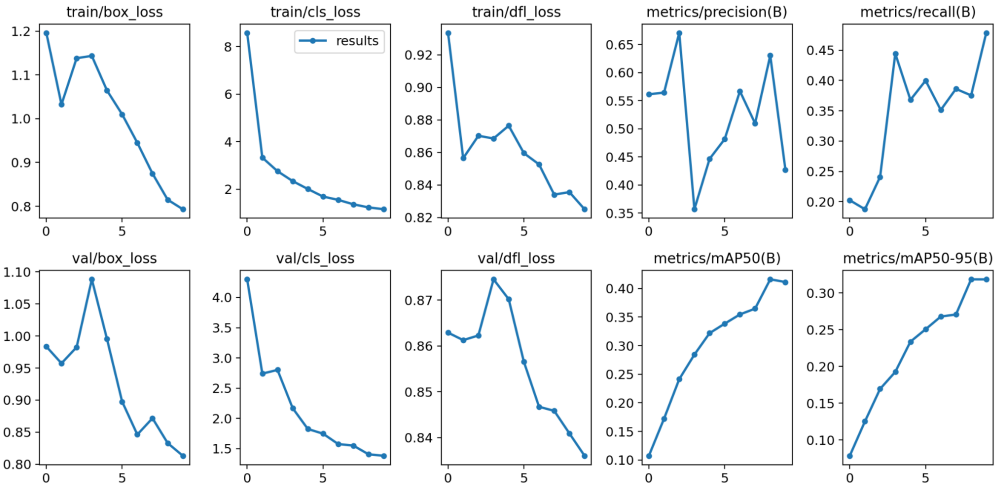
Uma das integrações do YOLOv8 que contribuiu com o projeto é com a aplicação Roboflow, onde é possível encontrar datasets, produzir datasets com mais facilidade, customizar e ajustar datasets, para variados projetos de inteligência artificial aplicados à detecção e reconhecimento de imagens.

Vários testes de treinamento foram feitos em datasets disponíveis no site, com resultados insuficientes. Uma solução encontrada foi o dataset GTSDB, também produzido pelo INI, autores do dataset GTSRB já utilizado no projeto. Um dos problemas com este dataset era que as anotações de localização e classe das placas presentes nas imagens foram fornecidas em um formato diferente do formato utilizado pelo YOLOv8, e assim precisaram ser convertidas por meio de um programa em python.

As anotações originais são disponibilizadas no formato x1, y1, x2, y2, para as coordenadas das caixas dos objetos, porém o YOLOv8 utiliza o formato centro x, centro y, largura, e altura. Após convertidas as anotações, as imagens foram carregadas no Roboflow, onde o conjunto de treinamento e validação foram separados, utilizando o padrão de 70% para treinamento e 30% para validação.

Os resultados do treinamento da rede se encontram abaixo, note para o gráfico de mAP50(B), que descreve a acurácia geral da rede, atingindo apenas 0.40, sendo um resultado fraco.

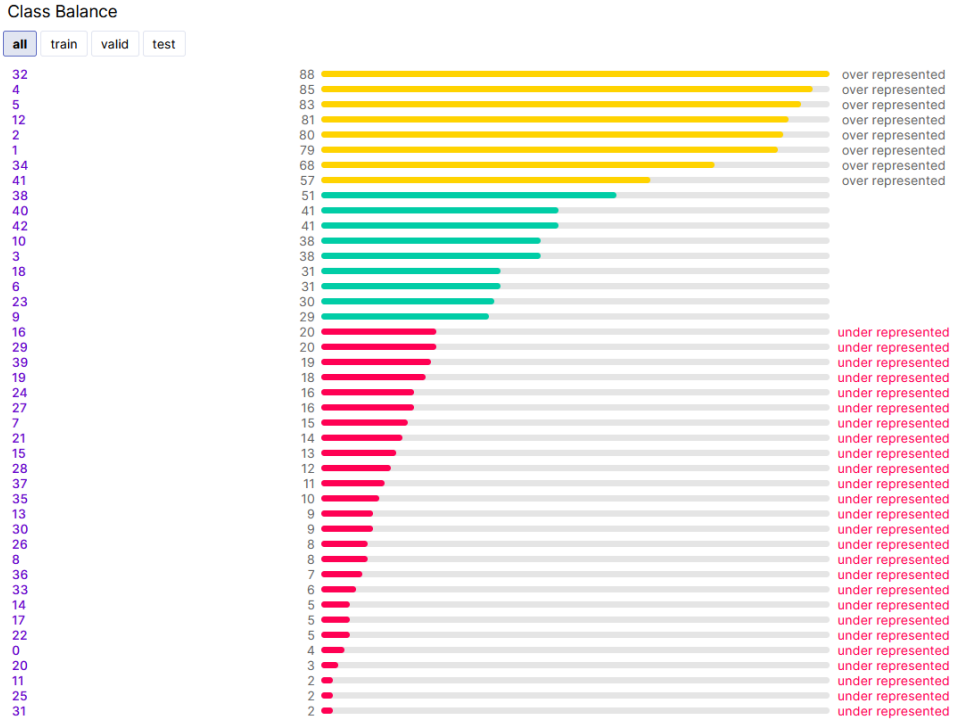
Figura 5: Gráficos de resultado iniciais da rede detectora



Fonte: elaborado pelo autor.

Devido a esses resultados iniciais foi feita uma análise do dataset. Este dataset se aparenta desbalanceado, com muitas classes de objetos pouco representadas. A Figura 6 mostra a frequência em que cada classe de objeto aparece em todo o dataset:

Figura 6: Distribuição do dataset



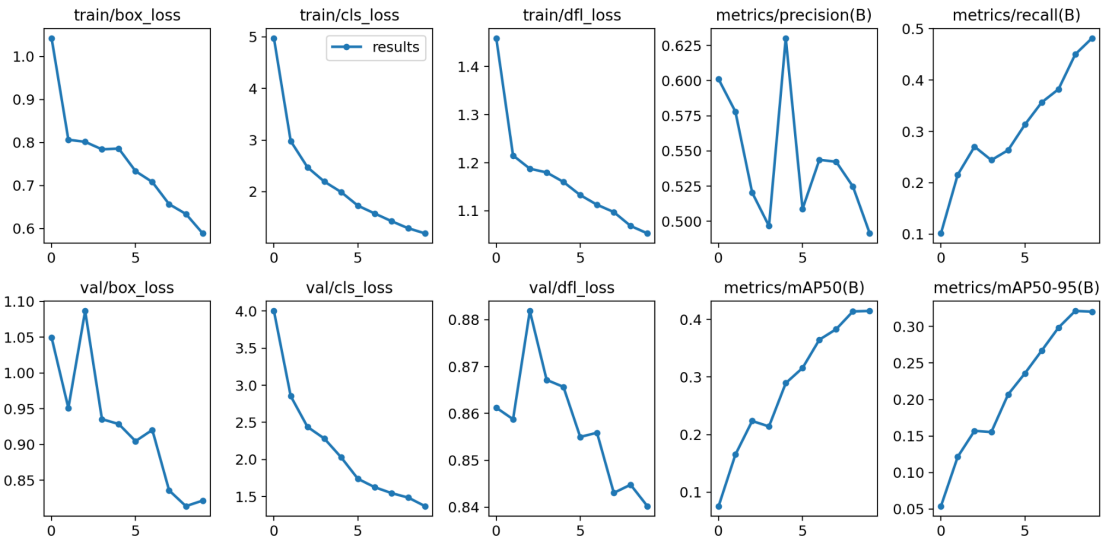
Fonte: elaborado pelo autor.

Um dos passos tomados para melhorar a performance de detecção da rede foi o aumento das imagens no dataset. Foram selecionadas 20 imagens de cada uma das 42 placas, do dataset GTSRB, onde se encontram apenas imagens

contendo somente a placa em si, e foram manualmente mapeadas suas regiões de interesse, ocupando quase a imagem toda. É importante notar que todas estas imagens foram utilizadas apenas no conjunto de treinamento, e não no conjunto de validação, pois o propósito do conjunto de validação é verificar se o modelo é capaz de detectar as placas em um cenário realístico, onde as placas estariam ocupando um pequeno espaço da imagem total.

A Figura 7 apresenta os gráficos dos resultados de treinamento, após essa mudança:

Figura 7: Resultados da rede detectora após a primeira mudança no dataset

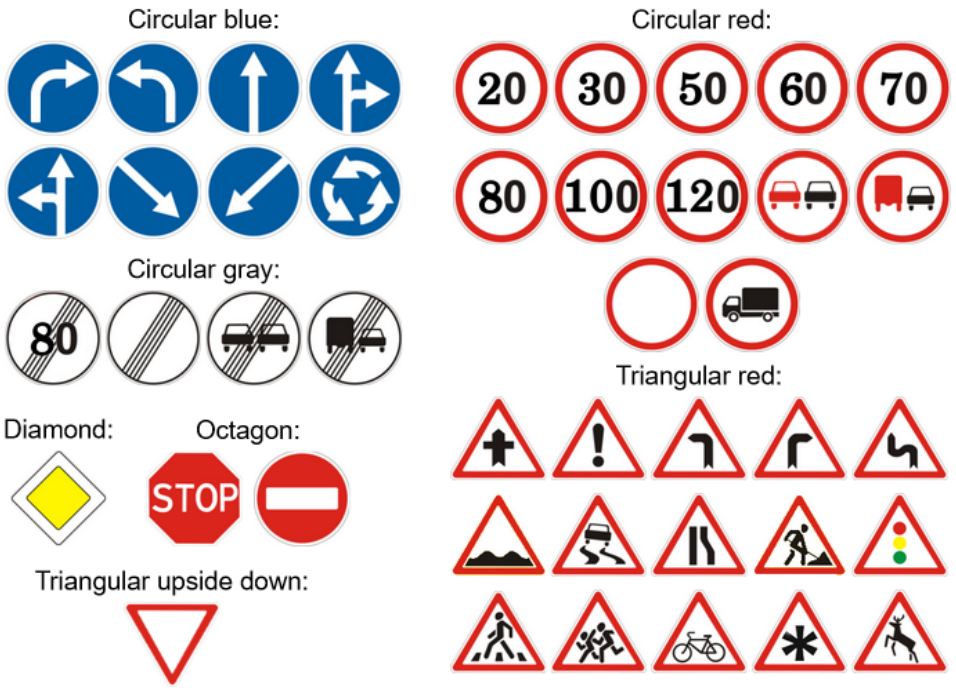


Fonte: elaborado pelo autor.

Ao observar os dois últimos gráficos da Figura 7, percebe-se que o ganho de acurácia não foi perceptível.

Outro passo que foi tomado em busca de melhor acurácia foi a redução do número total de classes, simplificando o processo de detecção da rede neural. As variadas placas foram agrupadas em classes por semelhança visual, produzindo um grupo menor de objetos para serem reconhecidos e detectados pela rede, para que fossem posteriormente classificados pela primeira rede desenvolvida neste projeto, uma rede especializada para este propósito. Na Figura 8 se encontra o agrupamento de classes que foi utilizado no projeto:

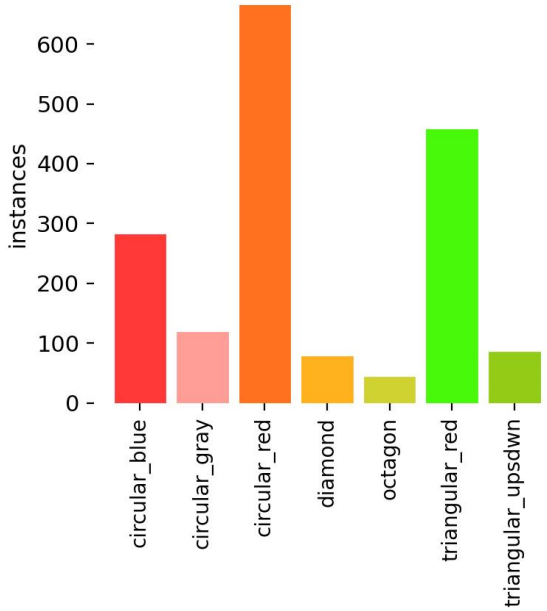
Figura 8: Agrupamento das placas em classes por semelhança visual



Fonte da imagem: elaborado pelo autor.

Após essa mudança, a rede trabalha apenas com 7 classes, e não com 42, trazendo grande simplificação no processo de detecção, além de trazer um pequeno balanceamento para o dataset, como visto na Figura 9:

Figura 9: Distribuição das classes por instância em imagem

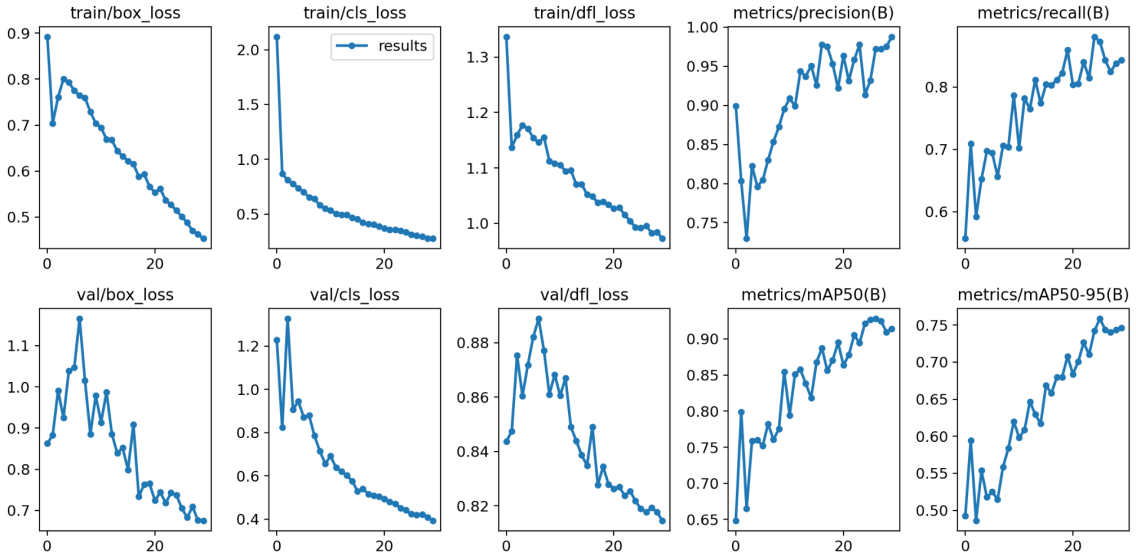


Fonte: elaborado pelo autor.

Esta segunda mudança trouxe um aumento expressivo na taxa de detecção, tornando mais factível que o conteúdo das caixas fosse exportado e processado pela

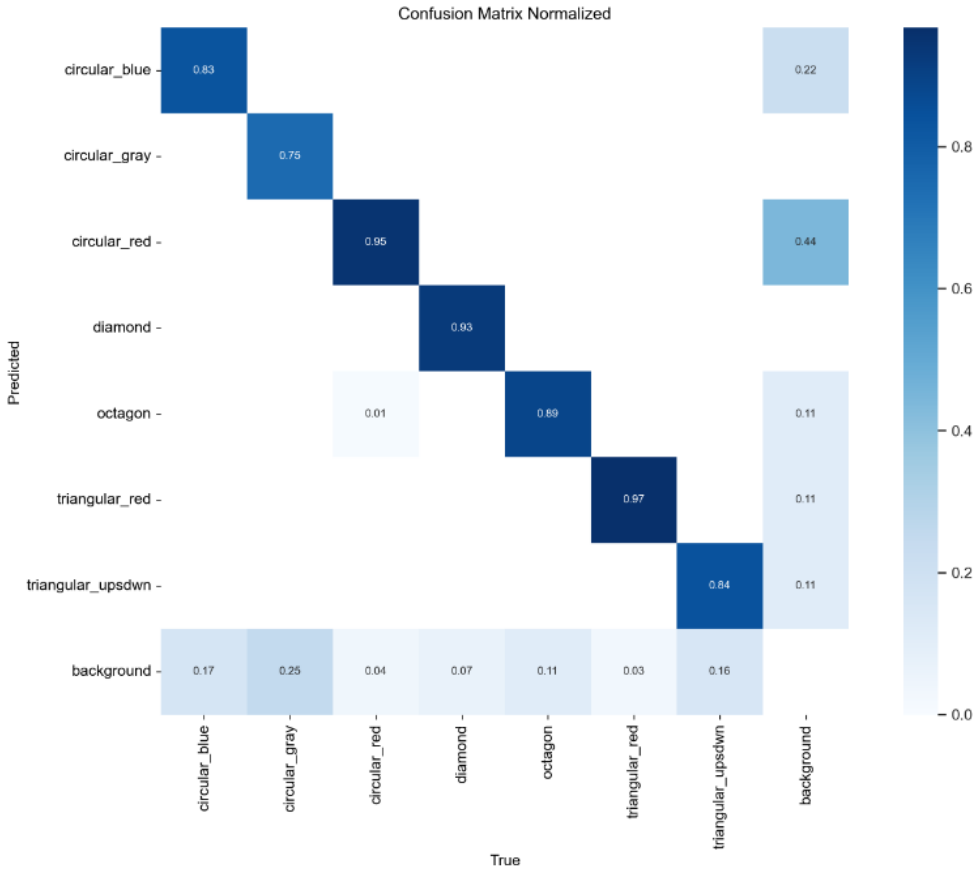
rede classificadora descrita no projeto. Na Figura 10 podemos observar no gráfico de mAP50(B) que a rede atingiu valores acima de 0.9 para a métrica mAP. Na Figura 11 é possível ver a matriz de confusão da rede classificadora, e a distribuição dos seus erros

Figura 10: Resultados da rede detectora após a segunda mudança no dataset



Fonte: elaborado pelo autor.

Figura 11: Matriz de confusão da rede detectora



Fonte: elaborado pelo autor.

Como apresentado na matriz de confusão da Figura 11, a maior parte dos erros se encontram na falha em encontrar e detectar uma placa, classificando as imagens como background, ou o erro de falso positivo, onde são reportadas detecções onde não há placas. Porém no quesito de classificação, a rede apresenta bons resultados, com alta acurácia de classificar as placas nos seus grupos devidos.

5. CONSIDERAÇÕES FINAIS

Notadamente o problema da detecção e reconhecimento automático da sinalização vertical de trânsito através do processamento de imagens não é uma tarefa simples. A quantidade de elementos que compõem o ambiente urbano, bem como a dinâmica de interações entre os participantes e o próprio movimento do observador implicam em uma tarefa complexa.

As técnicas de *Deep Learning*, em particular as redes neurais convolucionais (CNNs), têm mostrado bom desempenho em endereçar esse problema e conseguido lidar, simultaneamente, com a detecção de placas e a identificação (significado) do que elas regulamentam.

Entre os objetivos desta pesquisa estava o desenvolvimento de uma aplicação que fosse capaz de reconhecer a sinalização vertical de trânsito. Assim, a proposição de um modelo capaz de identificar e classificar diferentes placas foi parte essencial da aplicação. Como demonstrado experimentalmente, o programa tem a capacidade de detectar e classificar as placas, apresentando uma acurácia moderada em relação ao visto na literatura. Vale ressaltar que, embora trabalhando com um dataset constituídos de imagens reais, não foram realizados experimentos com câmeras embarcadas em um veículo em movimento.

A principal dificuldade do processo experimental do projeto é a qualidade dos dados necessários para o treinamento do modelo de inteligência artificial. Estudos futuros devem explorar o uso de outros datasets, e também o enriquecimento de dados inserindo ruído ou rotações nas imagens, para avaliações de desempenho e acurácia, avaliando os ganhos quando utilizando um conjunto de dados mais completo.

Nota-se que há oportunidades interessantes para pesquisa na área, inclusive no que diz respeito a uma abordagem diferenciada e que foi aplicada neste projeto, que consiste no uso de duas redes neurais trabalhando em conjunto para obter melhores resultados. A generalização dos tipos de placas trouxe um ganho de acurácia para a rede detectora, deixando detalhes como por exemplo a classificação

dos dígitos numéricos nas placas de limite de velocidade como uma tarefa para uma rede classificadora secundária.

6. REFERÊNCIAS

ETSI. Human Factors (HF); Intelligent Transport Systems (ITS); ICT in cars. ETSI, Abril de 2010. Disponível em: https://www.etsi.org/deliver/etsi_tr/102700_102799/102762/01.01.01_60/tr_102762v010101p.pdf. Acesso em 12, abril, 2023.

HAQUE, Wasif A; AREFIN, Samin; SHIHAVUDDIN, A.S.M; HASAN, Muhammad A. DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0957417420311283. Acesso em 9, abril, 2023.

NEWS, ONU. Acidentes de trânsito são a maior causa de morte de pessoas de 5 a 29 anos. Nações Unidas 21 de Novembro de 2021. Disponível em: https://news.un.org/pt/story/2021/11/1771092. Acesso em 12, abril, 2023.

REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI Ali. You Only Look Once: Unified, Real-Time Object Detection. Disponível em:https://arxiv.org/abs/1506.02640. Acesso em: 10, abril, 2023.

TERVEN, Juan R; ESPARAZA, Diana M. Cordova. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. Disponível em:https://arxiv.org/abs/2304.00501. Acesso em: 11, abril, 2023.

WHO. Road traffic injuries. World Health Organization, 20 de junho de 2022. Disponível em: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. Acesso em 12, abril, 2023.

Contatos: schenbergbruno@gmail.com e leandro.fernandes@mackenzie.br