



# Redes Neurais Convolucionais na detecção e classificação de placas de trânsito brasileiras

**João Pedro Knauer de Queiroz Verçosa**

**Proposta do Projeto Final de Graduação**

**Orientadora: Marley Maria Bernardes Rebuzzi Vellasco**

**Coorientador: Meyer Elias Nigri**

**CENTRO TÉCNICO CIENTÍFICO – CTC**

**DEPARTAMENTO DE INFORMÁTICA**

**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**Rio de Janeiro, Dezembro de 2022**

## Resumo

K. de Queiroz Verçosa, João Pedro. M. B. Rebuzzi Vellasco, Marley e Elias Nigri, Meyer. Redes Neurais Convolucionais na detecção e classificação de placas de trânsito brasileiras. Rio de Janeiro, 2022. 37p. Relatório de Projeto Final – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

O objetivo deste trabalho é estudar arquiteturas de Deep Learning, utilizando modelos de Redes Neurais Convolucionais para maximizar a capacidade de detecção e classificação de sinalização de trânsito em autoestradas. Além de comparar resultados entre modelos, pretende-se tornar os modelos capazes de reconhecer todas as placas nacionais brasileiras através do enriquecimento do conjunto de treino com imagens rotuladas geradas artificialmente.

Palavras-chave:

Deep Learning, Aprendizado Profundo, Redes Neurais, Redes Neurais Convolucionais, Classificação e Detecção de Imagens, Placas de Trânsito Brasileiras

## Abstract

K. de Queiroz Verçosa, João Pedro. M. B. Rebuzzi Vellasco, Marley e Elias Nigri, Meyer. Redes Neurais Convolucionais na detecção e classificação de placas de trânsito brasileiras. Rio de Janeiro, 2022. 37p. Proposta de Projeto Final – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

The objective of this paper is to study Deep Learning architectures using Convolutional Neural Networks models to maximize the ability to detect and classify traffic signs on highways. In addition to comparing results between models, it is intended to make the models capable of recognizing all Brazilian traffic plates by augmenting the training set with artificially generated labeled images.

Keywords:

Deep Learning, Neural Networks, Convolutional Neural Networks, Image Classification and Detection, Brazilian Traffic Signs

## SUMÁRIO:

<i>Sumário:</i> .....	1
<i>Lista de Imagens:</i> .....	2
<b>1 – INTRODUÇÃO</b> .....	<b>3</b>
1.1 – INTELIGÊNCIA ARTIFICIAL .....	3
1.2 – REDES NEURAS CONVOLUCIONAIS .....	3
1.3 – DETECÇÃO DE PLACAS DE TRÂNSITO .....	4
1.4 – AMBIENTE DE DESENVOLVIMENTO .....	4
- Ambiente Físico: .....	4
- Ambiente Cloud: .....	5
1.5 – CONHECIMENTOS APLICADOS .....	5
<b>2 – SITUAÇÃO ATUAL</b> .....	<b>6</b>
2.1 – AUSÊNCIA DE IMAGENS ANOTADAS COM PLACAS BRASILEIRAS .....	6
2.2 – PRINCIPAIS MODELOS DE REDES NEURAS CONVOLUCIONAIS .....	7
<b>3 – OBJETIVOS DO TRABALHO</b> .....	<b>8</b>
<b>4 – ATIVIDADES REALIZADAS</b> .....	<b>8</b>
4.1 – ESTUDOS .....	8
4.2 – MÉTODO E ATIVIDADES .....	9
<b>5 – IMPLEMENTAÇÃO</b> .....	<b>11</b>
5.1 – CONSTRUÇÃO DO CONJUNTO DE DADOS DE IMAGENS ARTIFICIAIS .....	11
5.2 – MODELOS ESCOLHIDOS .....	13
5.3 – MASK R-CNN .....	14
5.4 – YOLOV5 .....	16
5.5 – TREINAMENTOS .....	18
- Mask R-CNN: .....	18
- YOLOV5 .....	19
<b>6 – RESULTADOS</b> .....	<b>21</b>
6.1 – SIMPLEDATASET .....	22
6.2 – NORMALDATASET .....	23
6.3 – HUGEDATASET .....	23
<b>7 – CONSIDERAÇÕES FINAIS</b> .....	<b>23</b>
7.1 - CONCLUSÃO .....	23
7.2 – TRABALHOS FUTUROS .....	24
<b>8 – REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>27</b>
<b>9 – APÊNDICE</b> .....	<b>30</b>
TABELA 1 – LISTA, DISTRIBUIÇÃO E DESCRIÇÃO DAS CLASSES DAS MÁSCARAS DE PLACAS DE TRÂNSITO BRASILEIRAS .....	30
TABELA 2 – DISTRIBUIÇÃO DE CLASSES: SIMPLEDATASET .....	36
TABELA 3 – DISTRIBUIÇÃO DE CLASSES: NORMALDATASET .....	36

## LISTA DE IMAGENS:

Figura 1- Legenda Cronograma .....	10
Figura 2 - Imagens geradas artificialmente com sinalização de trânsito brasileira a partir do algoritmo proposto nesse projeto .....	12
Figura 3 - Distribuição de classes simpleDataset .....	13
Figura 4 - Distribuição de classes normalDataset .....	13
Figura 5 - Mask R-CNN , estrutura. Adaptado de [21] .....	15
Figura 6 - YOLOv5, estrutura. Adaptado de [21] .....	17
Figura 7 - Mask R-CNN, Gráfico da função de perda, simpleDataset, 90 épocas .....	18
Figura 8 - Mask R-CNN, Gráfico da função de perda, normalDataset, 70 épocas .....	18
Figura 9 - Mask R-CNN, Gráfico de perda no conjunto de validação, Azul: simpleDataset, Roxo: normalDataset .....	19
Figura 10 - YOLOv5, Gráfico da função de perda de BoundingBox, simpleDataset.....	19
Figura 11 - YOLOv5, Matriz de Confusão de treino, simpleDataset.....	20
Figura 12 - YOLOv5, Resumo de treinamento, simpleDataset.....	20
Figura 13 - YOLOv5, Gráfico da função de perda de BoundingBox, normalDataset.....	21
Figura 14 - YOLOv5, Resumo de treinamento, normalDataset.....	21
Figura 15 - Definição de precision e recall [24].....	21
Figura 16 - Definição IoU [24] .....	22
Figura 17 - Mask R-CNN, Detecções simpleDataset.....	24
Figura 18 - YOLOv5, Detecções normalDataset .....	24
Figura 19 - Roboflow, Mapa de calor, simpleDataset, classe A-24 .....	25
Figura 20 - Exemplos de placas da classe 'I-E – Sinalização Educativa' .....	25
Figura 21 - Diferenças entre templates digitais e templates gerados a partir de recortes reais.....	26
Figura 22 - Amostras da classe "I-IK .....	26

# 1 – Introdução

---

## 1.1 – Inteligência Artificial

A inteligência artificial é um assunto muito discutido no mundo da computação. O tema começou a tomar maiores proporções quando foi inventado o primeiro modelo teórico de rede neurais em meados do século XX e, também, quando surgiram as primeiras dúvidas teóricas sobre até que ponto é possível comparar a inteligência computacional à inteligência humana[1]. Desde então, o campo passou por diversas transformações, épocas de descrença e épocas de grande avanço científico.

Atualmente, inúmeros setores do mercado e muitas empresas fazem uso de algum tipo de Inteligência Artificial em seus programas, projetos ou processos. São aplicações em diversas áreas como robótica, reconhecimento de imagens, processamento de linguagem natural, planejamento, logística, predição de séries temporais, identificação de fraudes, entre outras.

Na última década, com a evolução do Deep Learning (Aprendizagem Profunda) e com a maior capacidade de processamento dos computadores, os modelos de Redes Neurais Artificiais vêm cada vez mais sendo empregados para encontrar soluções para grandes tarefas.

## 1.2 – Redes Neurais Convolucionais

Neste projeto são abordadas redes neurais convolucionais (*convolutional neural networks*), que são modelos de redes neurais artificiais especializados em reconhecimento de padrões em imagens. Existem tipicamente três tipos de camadas que fazem parte de uma rede neural convolucional (**CNN**): Camada Convolucional (*convolutional layer*), Camada de Junção (*pooling layer*) e a Camada Totalmente Conectada (*fully-connected layer*) [3].

Nas **camadas convolucionais**, a partir da entrada, são criados mapas de recursos. Eles funcionam como uma forma de a rede conseguir extrair características (*features*) dos dados de entrada, que podem ser importantes para a detecção de padrões pela rede. Esses mapas são montados a partir de uma convolução.

A convolução é o seguinte processo:

- (i) Cada neurônio dessa camada representa uma matriz (região) chamada de filtro (*kernel*);
- (ii) Uma *janela* é criada com a mesma dimensão que os filtros;
- (iii) Cada neurônio desliza sua janela pela imagem de entrada e aplica a cada iteração o seu filtro sobre uma região diferente, gerando um valor numérico final para cada região avaliada. Esse valor representa o quanto a característica desse filtro está presente na região da imagem, e
- (iv) Com os valores obtidos por cada neurônio para as regiões avaliadas, é gerado o mapa de ativação dessa característica[4][5].

Nas **camadas de junção**, chamadas de *pooling layers*, é realizada a redução das dimensões da rede. Dessa maneira, o número total de parâmetros que precisam ser treinados diminui[5].

Nas **camadas totalmente conectadas** existe uma arquitetura similar à das redes neurais artificiais, em que todos os neurônios são interconectados, e as

entradas alimentam os diferentes pesos que existem entre eles. A partir dos pesos que interligam esses neurônios, um deles é ativado na última camada, gerando a classificação que a rede neural deseja fazer[5].

### 1.3 – Detecção de Placas de Trânsito

Neste documento aborda-se a questão da detecção e do reconhecimento de placas de trânsito. Esse problema é estudado por engenheiros de diversas áreas em inúmeras aplicações, como sistemas avançados de auxílio de direção, sistemas de piloto automático de carros autônomos e semiautônomos, ou para auxiliar na manutenção de vias urbanas e autoestradas [2].

A maior parte dos projetos existentes com redes convolucionais utiliza conjuntos de placas de estradas europeias, como o *dataset* “German Traffic Sign Detection Benchmark” (GTSDb)[15] ou o BTS (“Belgian Traffic Signs”), mas o modelo desenvolvido neste projeto se utiliza das sinalizações brasileiras de trânsito.

Para isso, uma das fontes de dados utilizada neste trabalho vem de uma concessionária que administra uma rodovia em Curitiba, no Sul do Brasil. Os dados foram cedidos para o projeto pelo coorientador Meyer Nigri e consistem de uma série de vídeos de um trecho da rodovia BR-116, de 90 km de extensão (ida e volta), que sai de Tatuquara e segue contornando Quatro Barras até próximo do Portal da Graciosa.

A partir desses dados, em 2021, o ex-aluno Rodrigo Barcia, da PUC-Rio, rotulou placas de trânsito em 275 imagens retiradas desses vídeos e criou um conjunto anotado de placas de trânsito brasileiras [6]. Essa quantidade de dados é considerada insuficiente para os padrões de redes neurais convolucionais, mas através da transferência de aprendizagem e da utilização de modelos pré-treinados foram alcançados bons resultados.

O problema dessa fonte de vídeos disponível para o processo de criação de novas imagens é a restrição da amostragem a apenas um trecho da malha rodoviária brasileira, o que acaba não contemplando a totalidade de placas de sinalização existentes no código de trânsito brasileiro.

Volto a comentar sobre o projeto[6] do ex-aluno Rodrigo Barcia, no tópico “Construção do Conjunto de Dados com Imagens Artificiais”, do capítulo 5, no qual é proposta uma solução para a baixa quantidade de imagens rotuladas, solucionando o problema da pouca variabilidade de placas encontradas nesse trecho de autoestrada.

### 1.4 – Ambiente de Desenvolvimento

Dois ambientes de desenvolvimento foram utilizados para a construção e execução dos códigos necessários para a solução do problema. Por isso, essa seção também foi dividida em dois tópicos:

#### **- Ambiente Físico:**

Na maior parte do projeto foi utilizado um ambiente físico, com o sistema operacional Windows 10 Pro, com 16GB de RAM, CPU Intel i5-9400F com 6 núcleos e 2.90Ghz de frequência, com apoio de uma GPU NVIDIA GeForce RTX 3060 Ti. O código foi desenvolvido utilizando Notebooks rodando em *kernels* do Jupyter com Python 3.8.

Nesse ambiente, foi criada a ferramenta que gera imagens artificiais e engrandece o conjunto de imagens, processo que é abordado em detalhes na seção 1 do capítulo 5. Foram necessárias as funções da biblioteca OpenCV na versão 4.5, biblioteca essa que é desenvolvida em C e foi adaptada para Python. Através dela criaram-se máscaras e templates (recortes), o tamanho das imagens foi modificado, e assim foi possível colar templates em imagens e identificar os contornos de cada template.

O primeiro modelo de rede neural convolucional [18], que é abordado no capítulo 5, seção 3, também foi desenvolvido, executado e testado utilizando esse ambiente. Os códigos deste modelo podem ser consultados no GitHub [27].

#### - Ambiente Cloud:

Como segundo ambiente de desenvolvimento do projeto, foram utilizados ambientes na nuvem que têm a capacidade de realizar os treinamentos através de servidores. Isso melhora o consumo de recursos e possibilita estudos nos casos em que não existe hardware capaz de treinar os modelos convolucionais.

O segundo modelo [26] foi treinado utilizando GPUs disponibilizadas pelo GoogleColab na nuvem. O terceiro modelo é disponibilizado pela Roboflow [38], uma empresa privada que não especifica em qual ambiente acontece o processo de treino, mas também acontece em seus servidores.

Por fim, como foi utilizada a linguagem Python, existe um grande suporte da comunidade de código aberto da área de inteligência artificial, e por isso é possível ter acesso a bibliotecas como Scikit-Learn, Keras, PyTorch, Matplotlib e Numpy. Todas juntas forneceram as ferramentas necessárias para criar as imagens artificiais, estruturar os modelos convolucionais e analisar o desempenho desses modelos na tarefa proposta.

O código está disponível publicamente no GitHub [20]. Os conjuntos de dados e os arquivos dos modelos treinados podem ser acessados através do Google Drive [37].

### 1.5 – Conhecimentos Aplicados

No desenvolvimento do projeto foi possível ampliar os conhecimentos de computação gráfica ao criar o código capaz de gerar as imagens artificiais que foram agregadas ao conjunto de treino. Foi necessário realizar transformações, mudanças no intervalo das cores, recortes e colagens nas imagens.

Além da aplicação da lógica de programação para construção dos códigos e dos algoritmos, foi dada continuidade aos estudos na área de Inteligência Artificial e funcionamento de redes neurais. Mais especificamente com redes que trabalham com imagens, conhecidas como redes neurais convolucionais.

No capítulo 2, a seguir, é apresentado o contexto de pesquisa das redes neurais convolucionais, bem como os conceitos de redes detectores de um estágio e de redes detectores de dois estágios, e quais são as vantagens e desvantagens de cada um.

## 2 – Situação Atual

---

### 2.1 – Ausência de imagens anotadas com placas brasileiras

O treinamento de um modelo de rede neural convolucional requer um conjunto de dados utilizados para ajustar e calibrar seus parâmetros com o objetivo de permitir o reconhecimento de padrões. Especificamente as redes neurais convolucionais, por possuírem muitos parâmetros, necessitam de grandes conjuntos de dados.

Para que as imagens de um conjunto de dados possam ser lidas e interpretadas pela rede é necessário que elas sejam previamente rotuladas. Os rótulos indicam as localizações dos objetos a serem reconhecidos na imagem para o modelo. O modelo utiliza essas anotações para aprender e para validar suas inferências.

Existem grandes bancos públicos com enormes conjuntos de imagens rotuladas para treinamento de modelos na tarefa generalizada de reconhecer objetos em imagens, como o MS-COCO [13] e o Open Images Dataset [14]. Contudo, como neste trabalho o interesse está focado no reconhecimento de sinalização de trânsito, é preciso possuir imagens rotuladas com esse tipo de informação.

Há, também, conjuntos de dados rotulados especificamente para o reconhecimento e a classificação de placas de trânsito, como o German Traffic Sign Detection Benchmark [15], que trabalha com o reconhecimento e a classificação de placas de trânsito alemãs. Apesar de essas imagens estarem dentro do domínio do problema, não tratam de dados de placas de trânsito brasileiras, portanto, não ajudam no treinamento do modelo desenvolvido neste projeto.

Não existem grandes conjuntos de dados rotulados com sinalização de trânsito brasileira. O grande empecilho para a geração desses conjuntos é o fato de serem necessárias muitas horas de dedicação para desenvolvê-los manualmente. O processo envolve assistir a vídeos gravados em vias públicas brasileiras, selecionar *frames* com a informação de interesse (placas de trânsito), pegar cada uma das imagens selecionadas, rotular manualmente cada uma delas (desenhando o contorno de cada placa na imagem, com a especificação). Somente após todo esse trabalho de rotulação é possível apresentar os dados para o nosso modelo.

A tarefa de rotular as imagens manualmente conta com a ajuda de softwares que permitem ao usuário montar a máscara da placa na imagem. Isso é feito a partir da seleção de pontos que montam o contorno para o objeto de interesse. Porém, esse processo é lento. Para gerar grandes conjuntos de dados dessa forma, é preciso o esforço de toda uma comunidade ou de diversos voluntários, iniciativa que ainda não foi tomada no Brasil. Por isso, com poucos dados rotulados disponíveis os modelos atuais que reconhecem as placas de trânsito brasileiras têm dificuldade de treinamento.



## 2.2 – Principais modelos de Redes Neurais Convolucionais

Existem diversos tipos de modelos que implementam redes neurais convolucionais. Tradicionalmente é possível dividi-los em dois grupos: (i) detectores de dois estágios e (ii) detectores de um estágio [6].

- (i) Os modelos detectores de dois estágios precisam analisar as imagens duas vezes. Na primeira avaliação, é aplicada uma rede de propostas de regiões (*region proposal network*), e a imagem de entrada é dividida em regiões de interesse. Em seguida, a partir de cada região de interesse, é extraído um vetor de entrada. Esse vetor representa uma região da imagem que é avaliada pela segunda vez por uma rede convolucional, responsável por buscar e classificar os objetos presentes nas imagens em cada região selecionada.[31]
- (ii) Os modelos detectores de um estágio, a fim de buscar uma melhora de performance, utilizam apenas uma rede convolucional, que separa a imagem em uma grade de regiões e detecta os objetos em cada região [6][31].

Os modelos de dois estágios evoluíram a partir de 2014 através do *Regions with CNN features* (Regiões com características de redes neurais convolucionais) [6][7], chamada de *R-CNN* que alcança valores de  $mAP^{IoU=0.5} = 53,7\%$  no *dataset* PASCAL VOC [8], o que representa um aumento de 53% em relação ao modelo predecessor [7][9].

Em sequência, foram desenvolvidos os modelos: *Fast R-CNN* [16], *Faster R-CNN* [17] e *Mask R-CNN* [18][6]. Cada um deles utiliza recursos propostos pelos anteriores e apresentam melhoras de acurácia na detecção de objetos a partir de novas ideias. O *Mask R-CNN* (máscaras com regiões e redes neurais convolucionais) foi o modelo escolhido pelo ex-aluno de graduação da PUC-Rio em 2021 para fazer o primeiro projeto, utilizando o *dataset* privado da concessionária brasileira [6]. Foi alcançado  $mAP^{IoU=0.5} = 80,95\%$ , na tarefa da detecção de 17<sup>1</sup> classes diferentes de placas de trânsito, mesmo tendo um conjunto de dados de tamanho limitado.

Entre os modelos detectores de apenas um estágio, existem dois propostos em 2015: o *You Only Look Once-YOLO* (você só olha uma vez) [10] e o *Single Shot Multibox Detector-SSD* (detector de múltiplas caixas com uma captura) [11]. Ambos alcançaram boas taxas de reconhecimento de objetos, porém abaixo dos modelos de dois estágios. Por outro lado, modelos de apenas um estágio possuem a mais alta taxa de frames por segundo (FPS) quando executado em ambientes em tempo real [10][11].

---

<sup>1</sup> As classes 'MP1', 'MP2' e 'MP3' deveriam ser agrupadas em uma única classe 'MP' (Marcadores de Perigo), e as classes, 'R-19-H' e 'R-19-V', também deveriam ser agrupadas unicamente em R-19 (Regulamentação – velocidade máxima permitida), seguindo as regulamentações do CONTRAN (Conselho Nacional de Trânsito). Totalizando 14 classes regulares.

### 3 – Objetivos do trabalho

---

A proposta deste projeto consiste em:

- (i) estudar o funcionamento de modelos de redes neurais convolucionais;
- (ii) estudar soluções baseadas em redes neurais convolucionais para o problema de detecção e classificação de placas de trânsito;
- (iii) utilizar imagens geradas artificialmente para engrandecer e enriquecer o conjunto de dados de treino, suprimindo a falta de dados com imagens de sinalização de trânsito brasileira rotuladas;
- (iv) aumentar o número de classes que o modelo consegue reconhecer, para cobrir a maior variedade possível de placas de trânsito brasileiras;
- (v) implementar dois modelos de Redes Neurais Convolucionais para estudar o problema de detecção e classificação da sinalização de trânsito em autoestradas brasileiras;
- (vi) Comparar os resultados.

Este projeto tem seus códigos comentados, documentados e disponíveis no GitHub sob a licença GPLv3 [39]. A solução pode ser utilizada para identificação de placas de trânsito em imagens ou vídeos, utilizando os modelos treinados.

O problema da falta de grandes conjuntos de dados rotulados dentro do domínio é solucionado através de um algoritmo que gera imagens artificiais de treino já rotuladas. Essas imagens artificiais são adicionadas ao conjunto de imagens reais com objetivo de aumentá-lo significativamente, assim como melhorar a qualidade do mesmo, bem como a precisão do modelo. Esse algoritmo também foi capaz de aumentar a variedade e o número total de classes que compõem o conjunto de dados.

Os resultados de detecção e classificação desse projeto devem ajudar operações de empresas que necessitam da tecnologia, bem como permitir a avaliação de um modelo capaz de generalizar a tarefa de detectar e os tipos de placas de trânsito brasileiras.

### 4 – Atividades Realizadas

---

#### 4.1 – Estudos

Muitos artigos foram selecionados com o intuito de orientar os estudos nos temas de Redes Neurais Convolucionais [4], reconhecimento de imagem [7][8][9][10][11] e reconhecimento de placas de trânsito [2][3][6]. Além disso, dois livros foram utilizados nas pesquisas iniciais [1][5], contendo o conhecimento introdutório sobre Inteligência Artificial e Aprendizagem Profunda.

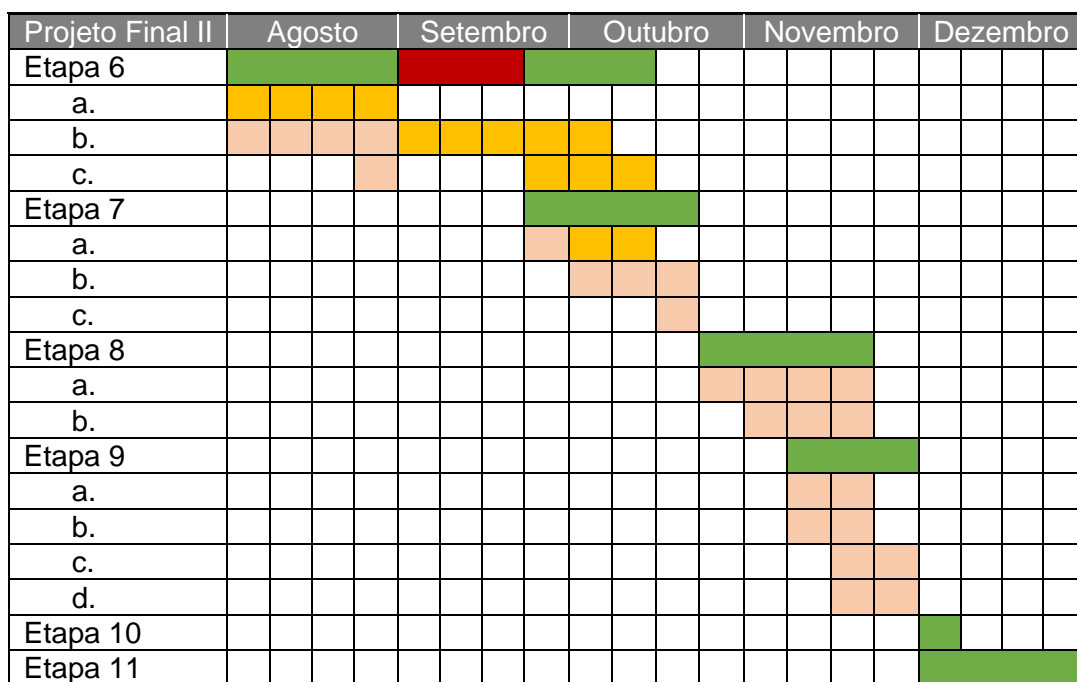
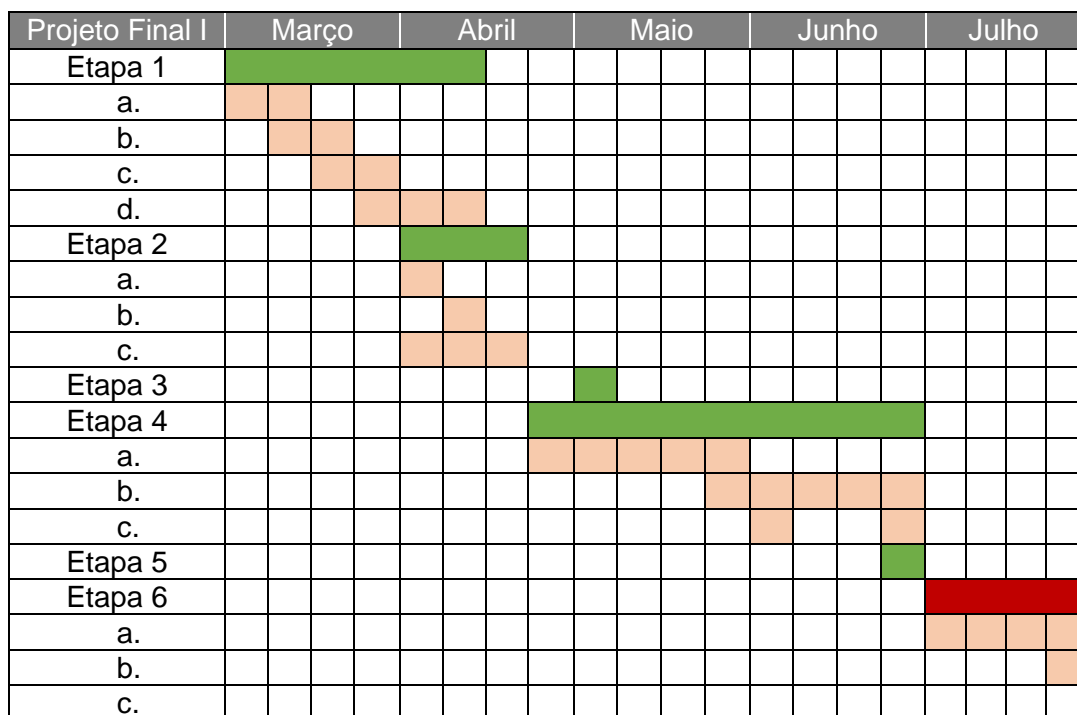
Os estudos de computação gráfica foram iniciados pelo artigo [16]. Em seguida, foram utilizados fóruns de dúvidas online e vídeos tutoriais para criar o código necessário, desenvolver o projeto e resolver os problemas que surgiram durante a implementação.

## 4.2 – Método e Atividades

Para o desenvolvimento do projeto, adotou-se uma metodologia ágil, baseada no Scrum [40].

O projeto foi dividido em etapas e foi realizado o planejamento de 11 atividades principais, sendo 3 delas referente a entregas de documentação oficial para a coordenação. A lista de atividades e o cronograma podem ser vistos a seguir:

- 1) Fazer um estudo sobre Redes Neurais Convolucionais:
  - a. Entender seu funcionamento,
  - b. as diferentes camadas,
  - c. as funções de cada camada,
  - d. registrar os estudos em um documento.
- 2) Estudar os modelos mais atuais de Redes Neurais Convolucionais.
  - a. Estudar modelos de dois estágios e modelos de um estágio.
  - b. Entender quais modelos melhor se adaptam para o problema de detecção e classificação de placas de trânsito tendo em vista conjunto de dados que será explorado no projeto.
  - c. Registrar os estudos e os modelos escolhidos para serem utilizados no projeto
- 3) Entrega de Proposta de Projeto Final I
- 4) Tratamento dos conjuntos de dados
  - a. Criar novos dados para auxiliar a tarefa de detecção de placas de trânsito brasileiras a partir do método proposto em [12].
  - b. Adicionar novas imagens rotuladas ao conjunto de dados de treino, do qual fazem parte as imagens da gravação cedida pela concessionária da rodovia de Curitiba
  - c. Registrar o avanço de criação das imagens nesse documento.
- 5) Entrega de Relatório de Projeto Final I
- 6) Implementar o primeiro modelo
  - a. Implementar o código necessário para a utilização do primeiro modelo: Mask-R-CNN
  - b. Rodar os treinamentos do modelo
  - c. Registrar a implementação e resultados do modelo e explicar seu funcionamento
- 7) Implementar o segundo modelo
  - a. Implementar o código necessário para a utilização do segundo modelo: YOLOv5
  - b. Rodar os treinamentos do segundo modelo
  - c. Registrar a implementação e resultados do modelo e explicar seu funcionamento
- 8) Comparar os modelos
  - a. Montar tabelas comparativa das métricas dos modelos.
  - b. Registrar nesse documento.
- 9) Efetuar revisão geral do projeto.
  - a. Revisão gramatical
  - b. Revisão Teórica
  - c. Revisão de Referências
  - d. Revisão de Formatação
- 10) Concluir o Relatório Final e a fazer a entrega do Relatório de Projeto Final
- 11) Preparar a apresentação para banca



Etapas:

Itens:

Feito

Planejado

Atrasado

Realizado

**Figura 1- Legenda Cronograma**

## 5 – Implementação

---

### 5.1 – Construção do Conjunto de Dados de Imagens Artificiais

Como o objetivo do projeto é treinar modelos que sejam capazes de reconhecer sinalização de trânsito brasileira, é necessário que existam dados rotulados para que a rede realize seu treinamento. Como descrito no capítulo 1, esses grandes conjuntos rotulados não existem para o contexto da sinalização de trânsito brasileiro, e é necessário um enorme esforço para gerar esse tipo de dado. Uma alternativa viável é criar esses conjuntos de dados artificialmente.

A inspiração para esse processo foi retirada do artigo [12]. Nessa publicação os autores conseguem melhorar a performance de um modelo de rede neural convolucional adicionando imagens novas geradas de uma maneira artificial ao conjunto de treino base.

Eles mostraram que os modelos treinados utilizando essas imagens adicionais conseguiam uma generalização melhor do problema, além de proporcionar melhor desempenho na tarefa de reconhecer e classificar placas de trânsito. O artigo explora modelos que foram avaliados no *German Traffic Sign Recognition Benchmark* [15].

Conforme o artigo [12], as imagens artificiais geradas foram incorporadas ao conjunto de treino e possuem um fundo fora do domínio do problema. As novas imagens têm como fundo: paisagens, animais, alimentos, cenas noturnas, pessoas caminhando, ou qualquer outro tipo de imagem, desde que não estejam representados nenhum tipo de sinalização de trânsito na imagem, veículos ou semáforos. É mais interessante que os fundos sejam de fora do domínio para evitar problemas da aparição de placas entre as imagens de treino que não sejam geradas pelo algoritmo e, portanto, não rotuladas.

O próximo passo envolve gerar máscaras (*templates* e pontos de contorno) para o maior número de placas possíveis seguindo as orientações de regulamentação do CONTRAN (Conselho Nacional de Trânsito). A partir desse ponto, o trabalho do algoritmo é montar as novas imagens de treino a partir dos fundos e dos *templates* selecionados, com as placas devidamente rotuladas.

As imagens geradas pelo algoritmo desse trabalho são semelhantes às originais do artigo [12], mas contêm sinalização de trânsito brasileira. As imagens de fundo foram selecionadas a partir do conjunto público de dados MS-COCO [13]. Foi escolhido o conjunto *train2017*, que contém mais de 120 mil amostras.

Através de uma API disponibilizada pelos detentores das imagens foi realizado um filtro e foram retiradas imagens que continham objetos das classes: 'bicycle', 'car', 'motorcycle', 'bus', 'train', 'truck', 'traffic light', 'fire hydrant', 'stop sign', 'parking meter' e 'clock'. Dessa forma, o total de imagens foi reduzido para cerca de 89 mil.



**Figura 2 - Imagens geradas artificialmente com sinalização de trânsito brasileira a partir do algoritmo proposto nesse projeto**

Após a execução do algoritmo que monta as colagens e rotula as imagens, foram gerados três conjuntos de dados:

- simpleDataset – com 382 imagens geradas artificialmente e 14 classes
- normalDataset – com 3.747 imagens geradas artificialmente e 242 classes
- hugeDataset – com 67.380 imagens geradas artificialmente

As imagens de treino receberam seus rótulos através do algoritmo enquanto eram geradas e esses novos conjuntos estão prontos para serem incorporados ao conjunto que contém 275 imagens reais rotuladas, disponibilizado pelo projeto [6].

Nesse momento também é demarcado o primeiro marco de evolução do projeto em relação ao estado da arte em contexto, foi possível expandir a capacidade de detecção do modelo de 14 classes para 242 classes, que podem ser conferidas na [‘Tabela 1 – Lista, distribuição e descrição das classes das máscaras de placas de trânsito brasileiras’](#) em anexo.

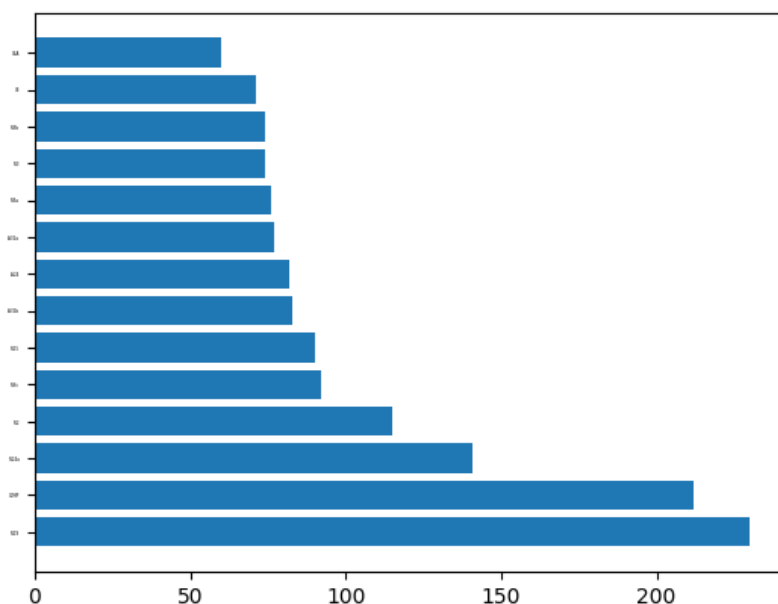
O *simpleDataset* representa um conjunto com um número menor de classes e foi avaliado e comparado com o projeto [6] para verificar se a adição de imagens artificialmente melhorou a precisão do modelo, como foi proposto por [12].

O *normalDataset* é uma seleção maior de imagens que contemplam todas as 242 classes. Nas avaliações desses modelos serão utilizadas imagens reais e artificiais, já que se fosse utilizado apenas o conjunto de imagens reais, nem todas as classes seriam contempladas pelos testes. Esse conjunto possui uma razão de cerca de dez imagens artificiais para cada imagem real.

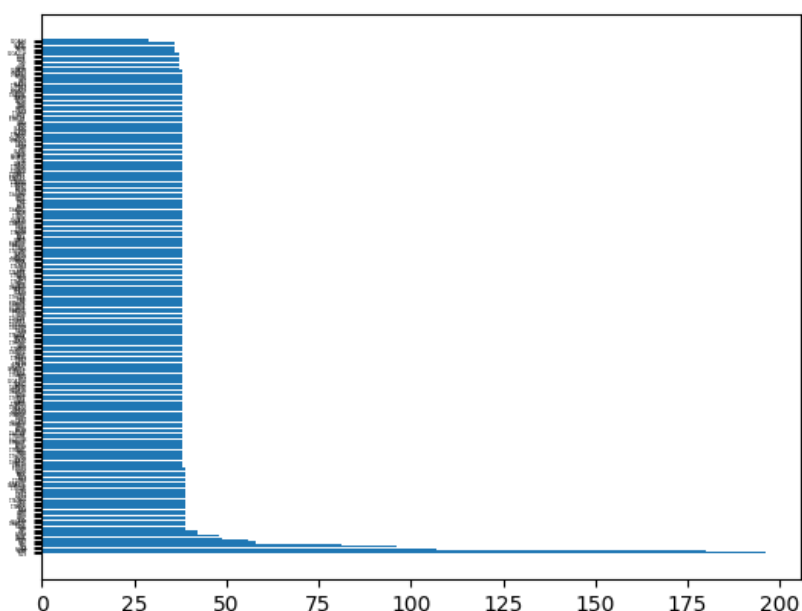
O *hugeDataset* é o maior conjunto que foi possível gerar usando as imagens dos dados do MS-COCO [13]. Com a adição das imagens reais, o conjunto apresenta uma razão de 245 imagens artificiais para cada imagem real, e isso pode gerar problemas no resultado do modelo, como *overfitting* e dificuldade de reconhecimento de padrões em novas imagens reais apresentadas ao modelo. O tempo para treinamento também é elevado devido ao número total de imagens. Nesse projeto, não foram realizados treinos no *hugeDataset*.

O código para a criação desses conjuntos de dados está disponível e comentado no GitHub nesse repositório [20].

No capítulo 9, Apêndice, também estão anexadas as tabelas: [‘Tabela 2 – Distribuição de classes: simpleDataset’](#), [‘Tabela 3 – Distribuição de classes: normalDataset’](#). Os gráficos abaixo refletem visualmente essas tabelas:



**Figura 3 - Distribuição de classes *simpleDataset***



O modelo YOLOv5 é um detector de um estágio que tem um treinamento mais leve e pode ser executado em ambientes em tempo real. Essa família de modelos demonstrou resultados piores inicialmente em relação aos detectores de dois estágios, mas melhorias e adaptações têm cada vez mais melhorado o potencial dessas redes.

A diferença de custo computacional e de tempo também fica exacerbada, por exemplo, pela quantidade de imagens que é enviada por lote durante o treino e pelo espaço físico ocupado na memória da placa de vídeo. Enquanto no modelo Mask R-CNN só é possível enviar até duas imagens por lote, sem estourar a memória disponível, no modelo YOLOv5 os lotes de imagens são compostos por 32 imagens.

Além desses dois modelos, foi utilizada uma ferramenta online chamada Roboflow [38]. Esse site possui uma plataforma que permite que você treine modelos na nuvem usando serviços de AutoML disponibilizados pela Amazon, Microsoft e Google. Infelizmente, nesse caso não sabemos qual estrutura de camadas foi utilizada, nem o modelo que foi utilizado no treino, porém a plataforma nos mostra a métrica dos resultados dos treinos sobre os conjuntos desejados, e também permite fazer inferência do modelo treinado direto pelo navegador.

Em todos os modelos foi empregue o princípio da transferência de aprendizagem. Ao invés de iniciar o treinamento do zero, com pesos aleatórios em cada neurônio, utilizamos modelos pré-treinados em grandes conjuntos de dados públicos. Apesar do programa de pré-treino ter sido diferente em cada caso, todos eles aproveitaram a distribuição de pesos de modelos previamente treinados no conjunto de dados MS-COCO [13].

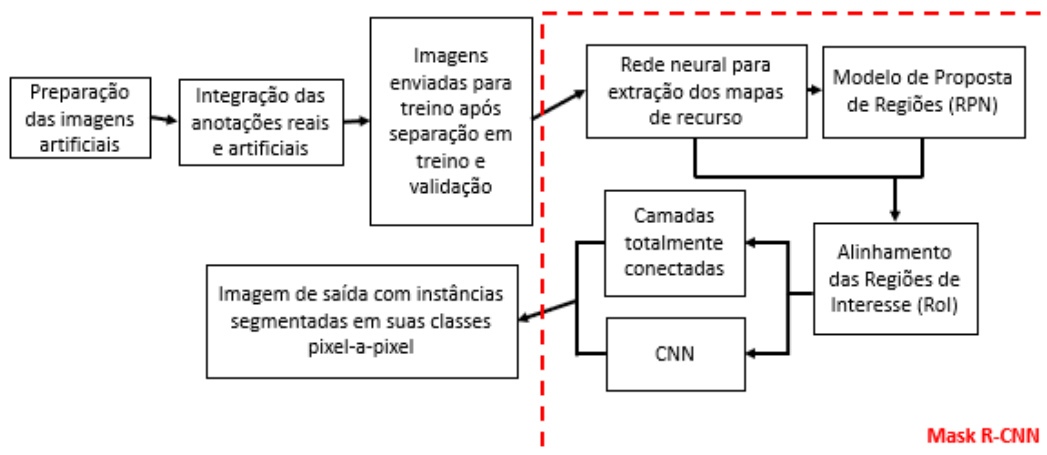
### 5.3 – Mask R-CNN

O modelo Mask R-CNN [18] é de segmentação de instâncias (*instance segmentation*) e se utiliza de algumas etapas para funcionar corretamente. É possível dividir essas etapas em três grandes processos que acontecem durante o treinamento. Ao treinar uma imagem, esse modelo:

- i) Prepara uma rede de proposta de regiões (*RPN – region proposal network*), que gera uma lista de possíveis regiões de interesse (*regions of interest – ROI's*)[23] e, através de uma rede piramidal de características (*FPN – feature pyramid network* [19][22]) são escolhidas as possíveis regiões que “devem” conter o(s) objeto(s) de interesse;
- ii) Para cada região de interesse selecionada na etapa anterior, o modelo aplica os processos convolucionais descritos no capítulo 1, que são comuns a Redes Neurais Convolucionais, e realiza suas inferências;
- iii) As previsões sobre a imagem de treino, são comparadas aos seus rótulos, e conforme a sua precisão ele reajusta seus pesos antes de seguir para próxima imagem de teste.

O modelo Mask R-CNN é uma evolução do Faster R-CNN e implementa uma nova capacidade na rede neural convolucional de segmentar as máscaras de cada objeto de interesse, de maneira que as máscaras fiquem delimitadas pixel-a-pixel do plano de fundo.





**Figura 5 - Mask R-CNN, estrutura. Adaptado de [21]**

O modelo Mask R-CNN possui 45 hiper parâmetros que podem ser configurados para otimização em diferentes problemas. Nos treinamentos no *simpleDataset* a parametrização foi reaproveitada a partir do projeto [6], o que vai permitir uma comparação dos resultados. A seguir uma lista dos principais parâmetros:

- **Images per GPU:** número de imagens por placa de vídeo. Esse parâmetro controla o tamanho do *batch* (grupo de imagens) que será alocado por vez para a memória da placa de vídeo. Mantemos esse parâmetro como 1, devido aos limites das GPUs e a alta definição das imagens.
- **Image Resize Mode:** modo de redimensionamento da imagem. Esse parâmetro configura como as imagens de treino serão redimensionadas para facilitar a alocação de memória no hardware. Os autores do modelo recomendam a configuração *square*, que ao receber uma imagem retangular como entrada, completa a menor dimensão com faixas pretas em ambas as margens, para tornar a imagem quadrada, e foi a configuração escolhida nos nossos treinos.
- **Image Max Dimension:** esse parâmetro controla o comportamento de redimensionamento das imagens, alterando-o conseguimos diminuir o tempo de treino e o consumo de memória do modelo. Optamos por redimensionar as imagens para 960x960.
- **RPN Anchor Scales:** esse parâmetro representa uma quintupla de potências de dois. O valor padrão é (32, 64, 128, 256, 512), mas para capturar objetos menores e diminuir o consumo de memória ele foi ajustado para (16, 32, 64, 128, 256). Esses valores representam as dimensões das regiões que serão buscadas na primeira etapa de treino do modelo pela RPN.
- **RPN Train Anchors Per Image:** número de regiões que serão propostas para cada imagem de treino. Por padrão são 256 regiões, mas esse valor foi reduzido para 200 para diminuir o número de regiões sem nenhuma detecção.
- **Train ROIs per Image:** esse hiper parâmetro controla a quantidade de regiões de interesse que vão ser enviadas para a segunda etapa

de treino, onde acontecem os processos convolucionais. Utilizamos os valores 100, 75 nos treinos.

- **Learning Rate:** Esse parâmetro controla a taxa de aprendizagem do modelo e foi implementada uma função que permite o controle desse valor durante o treinamento. Foi utilizado o programa de aprendizado proposto por [6], em que diminuimos o valor da taxa de aprendizagem durante a progressão das épocas.
- **Steps per Epoch:** número de passos tomados durante o treino para cada época. Por padrão utiliza-se o número de passos igual ao número de imagens de treino.
- **Validation Steps:** número de passos de validação tomados durante o treino ao final de cada época. Por padrão utiliza-se o número de passos igual ao número de imagens de validação. O número de imagens de validação é 15% do total de imagens no conjunto de treino para o *simpleDataset*, e 30% para os demais conjuntos.

Para o modelo Mask R-CNN utilizamos a transferência de aprendizagem a partir de um modelo pré-treinado no conjunto MS-COCO [13], que apresentava uma mAP = 0.6563 para tarefa proposta pelo conjunto.

## 5.4 – YOLOv5

Os códigos disponibilizados pela Ultralytics [26], são de Glenn Jocher e tratam de um projeto OpenSource [28], com mais de 40 contribuidores que incorpora a lógica do modelo YOLO (*You Only Look Once*) proposto por Joseph Redmon, Divvala Santosh, Ross Girshick e Ali Farhadi em 2015 [10].

Ao receber uma imagem o modelo cria um sistema de grades, onde cada célula é responsável por detectar um objeto, ou parte de um objeto, dentro de si. Com essa estratégia ele consegue alcançar boa acurácia e alta velocidade.

Esse sistema de grade evita que o modelo YOLOv5 precise de auxílio de uma rede de propostas de regiões, como no modelo Mask R-CNN, o que torna o treino mais rápido [29], mas ao mesmo tempo, dificulta o reconhecimento de objetos muito pequenos que ocorrem naturalmente em grupos [30].

O YOLOv5 é um detector de um estágio, e a rede pode ser dividida em três partes: *backbone*, *neck* e *head*. [31]

O *backbone* é responsável por extrair as características principais da imagem de entrada, e esse modelo utiliza uma rede de cruzamento parcial de estados (CSP – Cross Stage Partial Networks). O *neck* possui uma rede piramidal de características (*FPN*) que ajuda o modelo a performar bem e generalizar a tarefa para novas imagens. O *head* é responsável pela etapa final de detecção, ele aplica as inferências nas imagens e gera a saída final, com a imagem, suas classes e a confiança que ele tem em cada um dos casos.

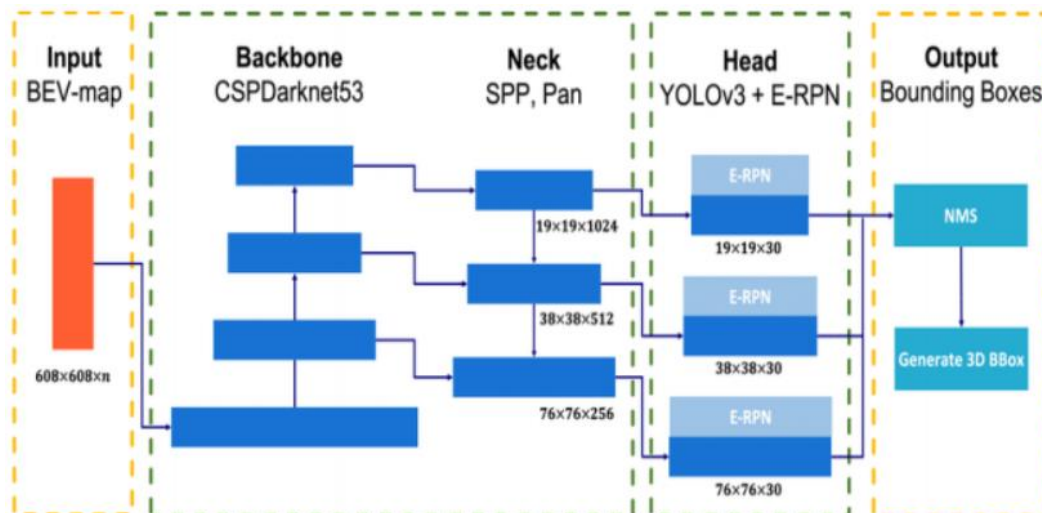


Figura 6 - YOLOv5, estrutura. Adaptado de [21]

Esse modelo utiliza como funções de ativação a *Leaky Relu* [35] e a *Sigmoid*. [31]. Para cálculo de perda esse modelo utiliza a *Binary Corss-Entropy with Logits Loss* [31] do framework *PyTorch*.

Nesse modelo existem 29 hiper parâmetros. Entre eles a maior parte lida com *data-augmentation* não foram utilizados para os treinos, ou seja, foram mantidos valores nulos. Além disso não foi desenvolvido uma parametrização específica para o problema, já que os resultados se mostraram acima das expectativas graças a transferência de aprendizagem. Entre os mais interessantes destacamos:

- **Learning-rate start:** Determina o tamanho do passo de aumento do learning-rate, o valor escolhido foi  $lr_0 = 0.01$ .
- **Learning-rate end:** valor final para o último ciclo de aumento do learning-rate. O valor utilizado foi  $lr_f = 0.2$ .
- **Momentum:** Valor responsável pela melhora do algoritmo de gradiente descendente do modelo [36], com valor de 0.937.
- **Mosaic:** Esse parâmetro determina se o modelo pega as imagens de treino e suas anotações e gera um mosaico na hora do treino e utiliza esse mosaico, como uma nova imagem de treino única[36]. Foi escolhido o valor 1.0, que significa que o programa está realizando essa técnica.
- **Weight Decay:** Valor que cria uma penalização a função de custo da rede, alterando a maneira que os pesos são comprimidos durante o processo propagação reversa[36]. O valor desse parâmetro para os treinos foi 0.0005.

## 5.5 – Treinamentos

### - Mask R-CNN:

No *simpleDataset* durante os treinos realizados com o modelo Mask R-CNN, existiu uma média de duração de 6 minutos por época e o modelo foi treinado por 90 épocas. Abaixo podemos observar os gráficos de perda do treinamento para esse caso:

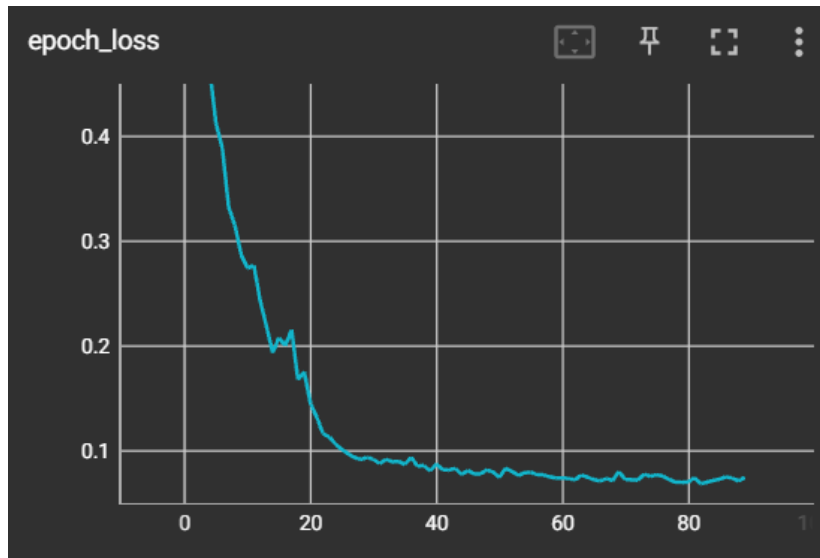


Figura 7 - Mask R-CNN, Gráfico da função de perda, *simpleDataset*, 90 épocas

O treino realizado no *normalDataset* com o modelo Mask R-CNN, ocorreu durante 70 épocas e a duração média de cada época foi de 30 minutos. Nesse conjunto temos mais de 4000 imagens e 242 classes diferentes para serem treinadas. Abaixo podemos observar o gráfico de perda:

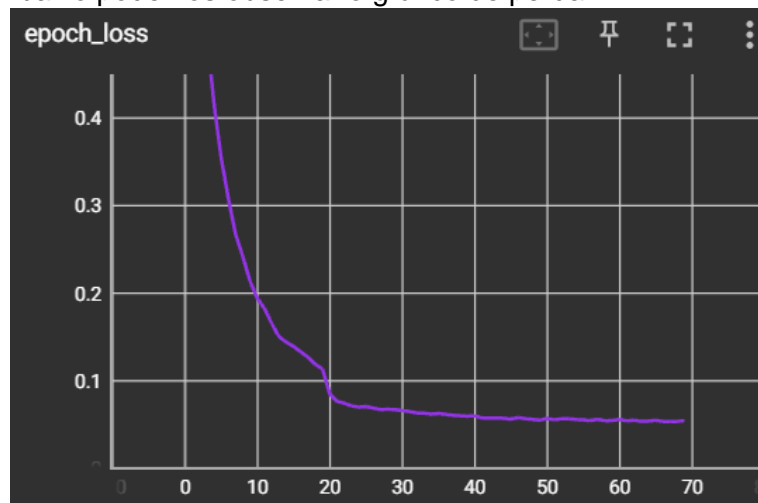
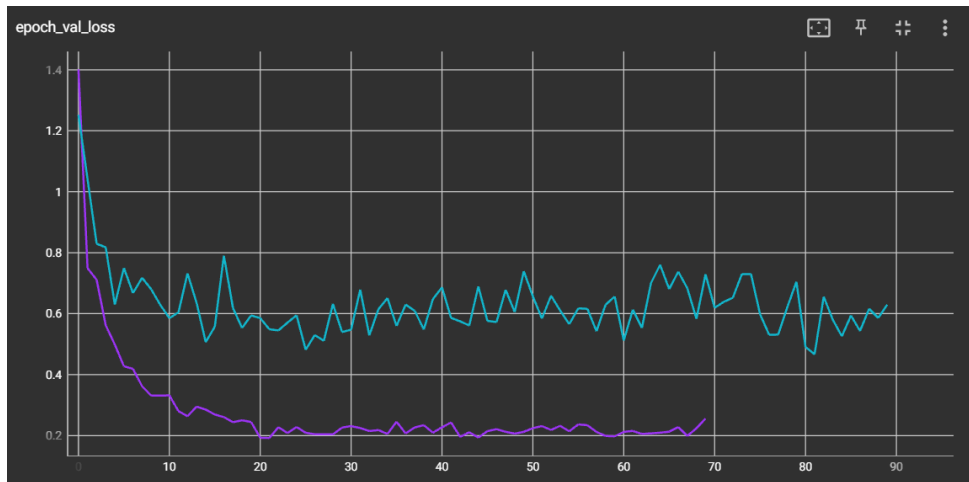


Figura 8 - Mask R-CNN, Gráfico da função de perda, *normalDataset*, 70 épocas

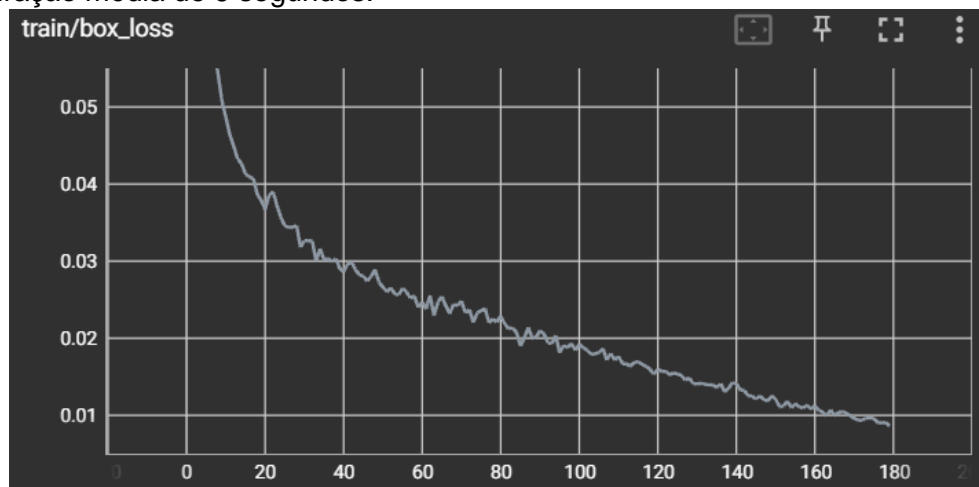
O modelo também disponibiliza gráficos da perda do conjunto de validação, aqui podemos ver, em azul, o gráfico de perda do modelo no *simpleDataset* e em roxo o gráfico de perda modelo no *normalDataset*.



**Figura 9 - Mask R-CNN, Gráfico de perda no conjunto de validação, Azul: simpleDataset, Roxo: normalDataset**

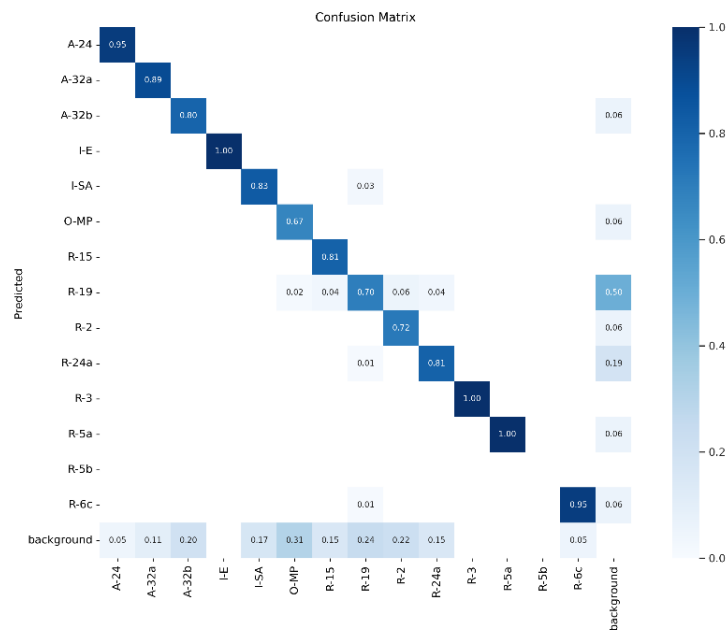
## - YOLOv5

No treino realizado no *simpleDataset* com o modelo YOLOv5 a transferência de aprendizagem foi iniciada a partir do *yolo5l*, que é uma das versões pré-treinadas disponibilizadas pela Ultralytics [26]. No caso, o modelo possui 46.5M parâmetros e foi pré-treinado no MS-COCO [13] durante 300 épocas e alcançou um  $mAP^{IoU=0.5} = 0.673$  na tarefa proposta pela Microsoft. Nesse projeto os treinos ocorreram durante 180 épocas e cada época apresentou duração média de 9 segundos.



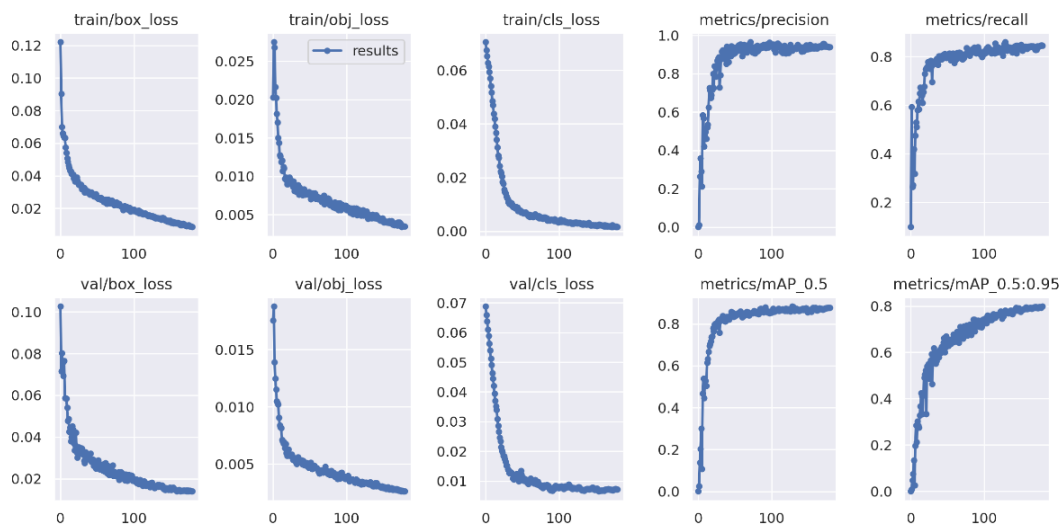
**Figura 10 - YOLOv5, Gráfico da função de perda de BoundingBox, simpleDataset**

Como temos apenas 14 classes sendo avaliadas nesse conjunto, é possível observar a matriz de confusão desse treinamento:



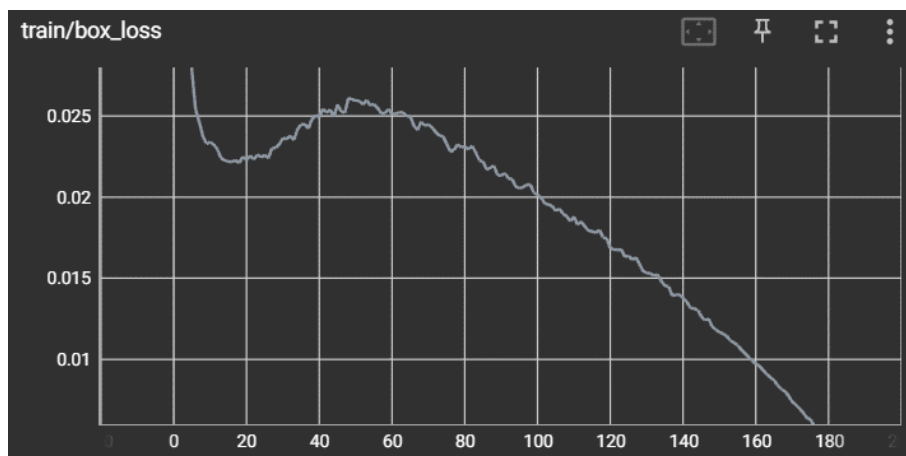
**Figura 11 - YOLOv5, Matriz de Confusão de treino, simpleDataset**

Por fim, temos o resumo de treino do YOLOv5 no *simpleDataset*:



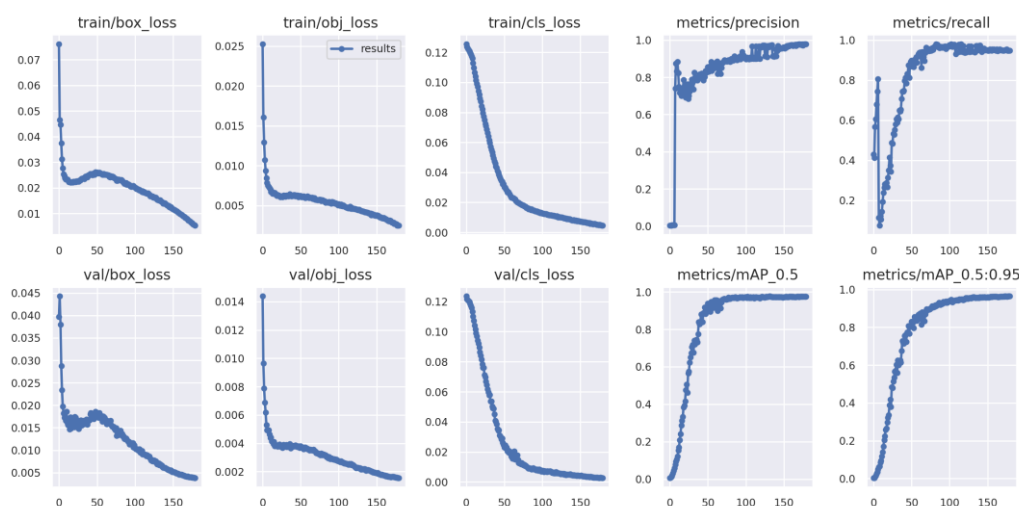
**Figura 12 - YOLOv5, Resumo de treinamento, simpleDataset**

No treino realizado no *normalDataset* com o modelo YOLOv5 foram realizadas 180 épocas, cada uma com duração média de 1 minuto. Além disso, através da transferência de aprendizagem, o modelo iniciou seus treinos a partir da distribuição de pesos do modelo pré-treinado *yolov5l*, como no caso anterior. A seguir podemos observar o gráfico de perda:



**Figura 13 - YOLOv5, Gráfico da função de perda de BoundingBox, normalDataset**

E abaixo podemos ver um resumo do treino em gráficos disponibilizados pelo modelo YOLOv5:



**Figura 14 - YOLOv5, Resumo de treinamento, normalDataset**

## 6 – Resultados

Os resultados são apresentados e divididos por *dataset*. A principal métrica para avaliação é mAP (*mean average precision*) com  $IoU = 0.5$  (*Intersection over Union*), que é muito comum para modelos de detecção de objeto e segmentação de instâncias. Ela é calculada pela média dos valores obtidos a partir da razão entre a área de interseção e a área de união, entre as predições do modelo e a rotulação original da imagem.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$TP$  = True positive

$TN$  = True negative

$FP$  = False positive

$FN$  = False negative

**Figura 15 - Definição de precision e recall [24]**



**Figura 16 - Definição IoU [24]**

A métrica  $mAP^{IoU=0.5}$  leva em consideração o nível de precisão (*precision*), que representa a proporção de identificações positivas corretas do modelo e a taxa de porcentagem de acertos (*recall*), que representa a proporção de verdadeiros positivos que foi identificada corretamente [24].

Utilizando o  $IoU = 0.5$  como um limite, define-se se as predições são Verdadeiros Positivos ( $TP$  – *true positive*) ou Falso Positivos ( $FP$  – *false positive*). Caso o resultado da  $IoU$  da predição do nosso modelo comparado ao que foi rotulado da imagem for maior que 0.5 consideramos um  $TP$ , e se for menor consideramos um  $FP$ [24].

É interessante observar que muitos modelos de redes neurais convolucionais publicam suas métricas relativas aos valores encontrados no conjunto de validação, e usam o conjunto de teste apenas para exemplos de inferência final do modelo. Seguindo os moldes da comunidade acadêmica, nesse projeto são apresentados os resultados da  $mAP^{IoU=0.5}$  aferidos durante o treino no conjunto de validação e, no capítulo 7, “Considerações Finais”, são expostos exemplos de detecções dos modelos nos seus respectivos conjuntos de teste.

Junto aos resultados dos modelos também são apresentados os dados relativos às métricas de Precision e Recall de cada modelo, quando disponível.

## 6.1 – simpleDataset

Nesse dataset é verificado se a técnica de engrandecer os dados com imagens artificiais realmente é capaz de melhorar a capacidade de detecção e classificação do modelo. Para isso mantemos o número de classes em 15 placas diferentes, o que contempla uma parcela muito pequena da variedade de placas existentes nas rodovias brasileiras. Em comparação com o modelo original[6]foi alcançado uma melhora de 3% utilizando o Mask R-CNN e 8,2% utilizando o YOLOv5.

Modelo	$mAP (IoU = 0.5)$	<i>Precision</i>	<i>Recall</i>
Mask R-CNN <sup>2</sup> [6]	0.8095	-	-
Mask R-CNN	0.834	-	-
YOLOv5	0.876	0.939	0.846
Modelo AutoML Roboflow	<b>0.888</b>	0.958	0.840

<sup>2</sup> Modelo treinado pelo aluno Rodrigo Garcia [6] no conjunto de dados contendo apenas imagens reais rotuladas manualmente.



## 6.2 – normalDataset

Com o *normalDataset* expandimos a diversidade de placas que os modelos conseguem analisar. Com o aumento do número de classes existe uma maior dificuldade para classificar as placas, já que diversas placas possuem formas e cores similares, porém também aumentamos o número de imagens totais de treino. Os códigos de nomeação seguem de acordo as orientações descritas pelos manuais do CONTRAN.

Modelo	<i>mAP (IoU = 0.5)</i>	<i>Precision</i>	<i>Recall</i>
Mask R-CNN	0.899	-	-
YOLOv5	<b>0.965</b>	0.980	0.948
Modelo AutoML Roboflow	0.964	0.853	0.946

## 6.3 – hugeDataset

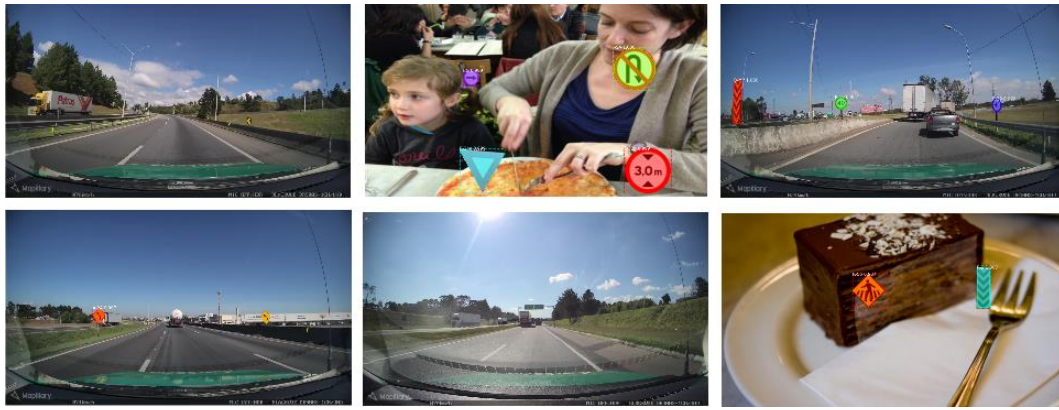
Devido ao tamanho do *hugeDataset*, que implicava em algumas horas de treino por geração com o modelo Mask R-CNN, e devido ao desequilíbrio entre a quantidade de imagens reais *versus* imagens artificiais, não foram realizados treinos no *hugeDataset*.

# 7 – Considerações Finais

## 7.1 - Conclusão

Como demonstrado acima, o trabalho foi capaz de provar os seus objetivos e de evoluir o processo de rotulação de imagens para contemplar uma variedade de 242 classes de placas diferentes. Os bancos de dados utilizados estão disponíveis em servidores na nuvem no Google Drive e podem ser acessados através das referências.

Além disso, foi possível analisar o comportamento do trabalho em três modelos diferentes e verificar o desempenho desses modelos na tarefa de reconhecimento de placas de trânsito brasileiras. Abaixo podemos conferir alguns exemplos de detecções no conjunto de teste para cada um dos modelos.



**Figura 17 - Mask R-CNN, Detecções simpleDataset**

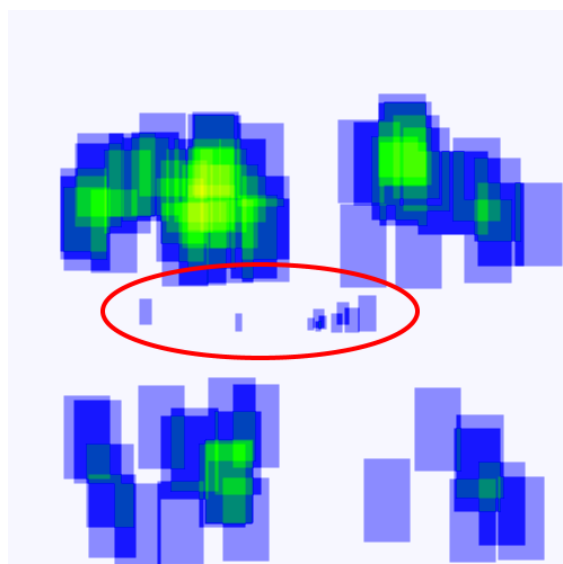


**Figura 18 - YOLOv5, Detecções normalDataset**

De qualquer forma, é possível apontar melhorias que devem ser capazes de aumentar ainda mais a precisão desses modelos. Nos conjuntos de testes, as métricas  $mAP^{IoU=0.5}$  no *normalDataset* foram de 0.851 para o modelo Mask R-CNN e de 0.905 para o modelo YOLOv5.

## 7.2 – Trabalhos Futuros

Ao trabalhar com imagens geradas artificialmente talvez seja necessário olhar os resultados com um pé atrás porque, afinal, o modelo está em parte sendo avaliado por imagens que foram geradas artificialmente, e não em imagens reais. Um exemplo disso pode ser observado a partir da análise do mapa de calor das anotações no *simpleDataset* para a classe A-24, que aparece um total de 82 vezes nesse conjunto:



**Figura 19 - Roboflow, Mapa de calor, simpleDataset, classe A-24**

No mapa de calor é possível observar 4 quadrantes nos quais existem predominâncias de amostras, que são as 4 regiões selecionadas pelo algoritmo que gera as imagens artificiais. Uma elipse vermelha foi adicionada a imagem através de um programa de edição para indicar um conjunto de amostras centrais separado dessas 4 regiões, que tem alta probabilidade de pertencer a anotações de imagens reais. Dessa forma, é possível imaginar que os modelos treinados nesse conjunto tenham dificuldade de generalizar a tarefa de detecção e classificação em situações reais dependendo da distância e da angulação da câmera em relação às placas na rodovia.

Também não foi realizado nenhum tipo de pré processamento de enriquecimento das imagens (*data augmentation*), como rotações, mudanças de escala das imagens ou alteração de brilho, contraste e saturação. Mas, para uma verdadeira melhora da qualidade das imagens artificiais, além dos ajustes de posicionamento sugeridos pelo mapa de calor e da utilização de engrandecimento de dados nas imagens em cada conjunto, é necessário aumentar as diferentes representações de *templates* de placa de trânsito disponíveis para esse algoritmo.

Na [Tabela 1 – Lista, distribuição e descrição das classes das máscaras de placas de trânsito brasileiras](#) é possível observar que algumas classes têm mais de uma amostra. Isso acontece porque existe mais de uma placa daquele tipo de classe, como é o caso da classe “I-E – Sinalização Educativa” que tem 25 amostras, todas da mesma classe, mas com diferenças entre si:

NA DÚVIDA NÃO ULTRAPASSE	SOB NEBLINA NUNCA ULTRAPASSE
EVITE DANOS À SINALIZAÇÃO	RESPEITE A VIDA
USE O CINTO DE SEGURANÇA	OBEDEÇA A SINALIZAÇÃO

**Figura 20 - Exemplos de placas da classe ‘I-E – Sinalização Educativa’**

Porém, a maioria das classes possui apenas uma amostra relacionada, e para conseguir realizar a expansão de 15 classes para 242 a maioria dessas novas classes adicionadas utiliza apenas um *template* gerado a partir de uma imagem digital, e não de um recorte de uma situação real. Abaixo podemos observar algumas diferenças entre imagens digitais e recortes retirados de imagens reais, usando como exemplo as amostras da classe “O-MP – Marcador de Perigo”:



**Figura 21 - Diferenças entre templates digitais e templates gerados a partir de recortes reais**

Os fundos laranjas existem para que o programa seja capaz de identificar corretamente o contorno dessas placas e de realizar a colagem nas imagens artificiais que ele deseja criar. Em relação a questão das diferenças entre esses dois tipos de *templates*, é possível apontar alteração nas cores das placas, no formato e na textura dessas imagens.

Como destacado, muitas classes têm apenas uma amostra digital como opção de escolha para o algoritmo que gera as imagens artificiais e, portanto, as imagens resultantes podem não conseguir generalizar as situações que ele vai encontrar em imagens reais. Não existiam imagens reais suficientes no banco de dados disponível no início do projeto que contemplassem todas as 242 classes, razão pela qual essa abordagem foi necessária.

Além disso, existe uma frequência real em que essas placas aparecem no dia a dia de aplicação do modelo, como por exemplo placas de Advertência e Indicação, que são muito mais frequentes que placas de Obras e de Turismos e Serviços Auxiliares. Então, para criar um modelo que tenha uma melhor performance nessas placas mais frequentes, podem ser adicionadas novas amostras retiradas de imagens reais, aumentando a variabilidade das imagens geradas artificialmente.

Um exemplo claro disso é a classe “Indicação – Indicador Quilométrico”, que possui apenas duas amostras, mas aparece pelo menos uma vez por quilômetro durante um percurso em uma estrada e sempre contendo informação diferente.



**Figura 22 - Amostras da classe "I-IK"**

Outra possibilidade, que contrapõe a ideia original que inspirou parte deste projeto [12], é adicionar fundos que fazem parte do domínio do problema, mas garantindo que não existam placas não rotuladas nesses fundos, e aplicar o algoritmo de geração artificial utilizando as mesmas imagens de fundo. No banco de dados disponibilizado pela concessionária da rodovia de Curitiba existem diversos *frames* de estrada que não possuem nenhuma sinalização de trânsito no fundo, e essas são imagens de alta qualidade, que podem servir de fundo para novas imagens geradas artificialmente a partir das máscaras.

Por fim, quando as imagens geradas artificialmente estiverem com um grau de semelhança mais próximo ao real, com a utilização de modelos que implementem processos de mecanismos de atenção, como os modelos propostos em [32][33][34], os níveis de detecção e classificação devem melhorar significativamente, possibilitando a criação de aplicações reais que utilizem esses modelos.

Mais ousado ainda seria utilizar as imagens geradas artificialmente e tentar desenvolver Redes Generativas Adaptativas (GANs) que conseguissem propagar e gerar novas imagens semelhantes para popular com ainda mais amostras os bancos de dados. Não obstante, mantendo um alerta ligado ao analisar os resultados e lembrar que, ao trabalhar com imagens artificiais, são adicionados novos dados ao modelo que devem trazer benefícios e não enviesamento.

## 8 – Referências bibliográficas

---

- [1] BODEN, Margaret A.. **Artificial Intelligence**. California, Academic Press Inc, 1996. 15 p. Disponível em: <https://books.google.com.br/books?hl=en&lr=&id=ixmRIL9jclC> Acesso em: 11/04/2022.
- [2] GARCÍA, Alvaro Arcos, GARCÍA, Juan A. Alvares, MORILLO, Luís M. Soria. **Evaluation of Deep Neural Networks for traffic sign detection systems**. Neurocomputing (2018), Doi: <https://doi.org/10.1016/j.neucom.2018.08.009> Disponível em: <https://sci-hub.hkvisa.net/10.1016/j.neucom.2018.08.009#> Acesso em: 13/04/2020
- [3] KUMAR, Amara Dinesh, KARTHIKA, R., PARAMESWARAM, Latha. **Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks**. International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018. Disponível em: <https://arxiv.org/ftp/arxiv/papers/1805/1805.04424.pdf> Acesso em: 13/04/2022
- [4] O'SHEA, Keiron, NASH, Ryan. **An Introduction to Convolutional Neural Networks**. 2015. Disponível em: <https://arxiv.org/pdf/1511.08458.pdf> Acesso em: 07/04/2022
- [5] Data Science Academy. **Deep Learning Book**. 2022. Disponível em: <https://www.deeplearningbook.com.br/> . Acesso em: 18/03/2022
- [6] BARCIA, Rodrigo Lema, **Automating the recognition of traffic signs on roads using artificial neural network**. 2021. Doi: <https://doi.org/10.17771/PUCRio.acad.56926>
- [7] GIRSHICK, Ross, DONAHUE, Jeff, DARREL, Trevor, MALIK, Jitendra. **Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)**. 22 de Outubro de 2014. Disponível em: <https://arxiv.org/pdf/1311.2524.pdf> . Acesso em: 17/04/2022



- [8] EVERINGHAM, Mark, VAN GOOL, Luc, WILLIAMS, Christopher K. I., WINN, John, ZISSERMAN, Andrew. **The PASCAL Visual Object Classes (VOC) Challenge**. IJCV. 2010. Disponível em: [https://homepages.inf.ed.ac.uk/ckiw/postscript/ijcv\\_voc09.pdf](https://homepages.inf.ed.ac.uk/ckiw/postscript/ijcv_voc09.pdf) . Acesso em: 17/04/2022
- [9] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders. **Selective search for object recognition**. IJCV, 2013. Disponível em: <https://sci-hub.hkvisa.net/10.1007/s11263-013-0620-5> Acesso em: 17/04/2022
- [10] REDMON, Joseph, SANTOSH, Divvala, GIRSHICK, Ross, FARHADI, Ali. **You Only Look Once: Unified, Real-Time Object Detection**. CoRR. 2015. Disponível em: <http://arxiv.org/abs/1506.02640.pdf> Acessado em: 21/04/2022
- [11] LIU, Wei, ANGUELOV, Dragomir, ERHAN, Dumitru, SZEGEDY, Christian, REED, Scott, FU, Cheng-Yang, BERG, Alexander C.. **SSD: Single Shot Multibox Detector**. CoRR. 2015. Disponível em: <http://arxiv.org/abs/1512.02325.pdf> Acessado em: 21/04/2022
- [12] TORRES, Lucas Tabelini, PAIXÃO, Thiago M., BERRIEL, Rodrigo F., de SOUZA, Alberto F., BADUE, Claudine, SEBE, Nicu, OLIVEIRA, Thiago Santos. **Effortless Deep Training for Traffic Sign Detection Using Templates and Arbitrary Natural Images**. CoRR. 2019. Disponível em: <https://arxiv.org/pdf/1907.09679v1.pdf> Acessado em: 14/04/2022
- [13] LIN, Tsung-Yi, MAIRE, Michael, BELONGIE, Serge, BOURDEV, Lubomir, GIRSHICK, Ross, HAYS, James, PERONA, Pietro, RAMANAN, Deva, ZITNICK, C. Lawrence, DOLLÁR, Piotr. **Microsoft COCO: Common Objects in Context**. CoRR. 2015. Disponível em: <https://arxiv.org/pdf/1405.0312.pdf> . Acessado em: 23/06/2022
- [14] KUZNETSOVA, Alina, ROM, Hassan, ALLDRIN, Neil, UIJLINGS, Jasper, KRASIN, Ivan, PONT-TUSET, Jordi, KAMALI, Shahab, POPOV, Stefan, MALLOCI, Matteo, KOLESNIKOV, Alexander, DUERIG, Tom, FERRARI, Vittorio. **The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale**. CoRR, 2020. Disponível em: <https://arxiv.org/pdf/1811.00982.pdf> . Acessado em: 23/06/2022.
- [15] STALLKAMP, Johannes, SCHILIPSING, Marc, SALMEN, Jan, IGEL, Christian. **The German Traffic Sign Recognition Benchmark: A multi-class classification competition**. IEE International Joint Conference on Neural Networks, 2011. Disponível em: [https://benchmark.ini.rub.de/gtsrb\\_about.html](https://benchmark.ini.rub.de/gtsrb_about.html). Acessado em: 23/06/2022.
- [16] GIRSHICK, Ross, **Fast R-CNN**. IEEE International ICCV 2015. Acessado em: 31/07/2022. Disponível em: [https://openaccess.thecvf.com/content\\_iccv\\_2015/papers/Girshick\\_Fast\\_R-CNN\\_ICCV\\_2015\\_paper.pdf](https://openaccess.thecvf.com/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf)
- [17] GIRSHICK, Ross, SUN, Jian, HE, Kaiming, REN, Shaoqing. **Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks**. Advances in Neural Information Processing Systems 28 (NIPS 2015). Acessado em: 31/07/2022. Disponível em: <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf>
- [18] GIRSHICK, Ross, DOLLÁR, Piotr, GKIOXARI, Georgia, HE, Kaiming. **Mask R-CNN**. IEEE International ICCV 2017, pp. 2961-2969. Acessado em: 31/07/2022. Disponível em:

- [https://openaccess.thecvf.com/content\\_ICCV\\_2017/papers/He\\_Mask\\_R-CNN\\_ICCV\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf)
- [19] LIN, Tsung-Yi, DOLLÁR, Piotr, GIRSHICK, Ross, HE, Kaiming, HARIHARAN, Bharath, BELONGIE, Serge. **Feature Pyramid Networks for Object Detection**. CoRR. Acessado em: 08/10/2022. Disponível em: <https://arxiv.org/pdf/1612.03144.pdf>
  - [20] VERCOSA, João Pedro Knauer de Q. **Final-Thesis-Dataset**. GitHub. Disponível em: <https://github.com/JPVercosa/Final-Thesis-Dataset>
  - [21] MYLVAGANAM, Pravina, DISSANAYAKE, Maheshi B. **Deep Learning for Arbitrary-Shaped Water Pooling Region Detection on Aerial Images**. Conference Paper, Julho, 2022. Acessado em 12/10/2022. Disponível em: <https://ieeexplore.ieee.org/document/9906204>
  - [22] ZHANG, Xiang. **Simple Understanding of Mask RCNN**. Medium Article, 2018. Acessado em: 08/10/2022. Disponível em: <https://alittlepain833.medium.com/simple-understanding-of-mask-rcnn-134b5b330e95>
  - [23] KARMARKAR, Tanay. **Region Proposal Network (RPN) – Backbone of Faster R-CNN**. Medium Article, 2018. Acessado em 08/10/2022. Disponível em: <https://medium.com/egen/region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9>
  - [24] HUI, Jonathan. **mAP (mean Average Precision) for object Detection**. Medium Article, 2018. Acessado em: 02/10/2022. Disponível em: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>
  - [25] KELLY, Adam. **Mask\_RCNN**. Github. Disponível em: [https://github.com/akTwelve/Mask\\_RCNN](https://github.com/akTwelve/Mask_RCNN). Acessado em: 27/08/2022
  - [26] ULTRALYTICS. **YOLOv5**. Github. Disponível em: <https://github.com/ultralytics/yolov5>. Acessado em: 07/10/2022
  - [27] VERCOSA, João Pedro Knauer de Q. **Final-Thesis-Mask-RCNN**. Github. Disponível em: <https://github.com/JPVercosa/Final-Thesis-Mask-RCNN>
  - [28] JOCHER, Glenn and 40 others. **YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations**. Disponível em: <https://zenodo.org/record/7002879#.Y1RRr3bMJPY>. Acessado em: 22/10/2022
  - [29] ULTRALYTICS. **YOLOv5 Documentation**. Disponível em: <https://docs.ultralytics.com/> Acessado em: 23/10/2022
  - [30] BANDYOPADHYAY, Hmrishav. **YOLO: Real-Time Object Detection Explained**. V7Labs. Disponível em: <https://www.v7labs.com/blog/yolo-object>. Acessado em: 23/10/2022
  - [31] RAJPUT, Mihir. **YOLOv5 – Explained and Demystified**. TowardsAI. Disponível em: <https://towardsai.net/p/computer-vision/yolo-v5>. Acessado em 23/10/2022
  - [32] GUO, Meng-Hao, XU, Tian-Xing, LIU, Jiang-Jiang, LIU, Zheng-Ning, JIANG, Peng-tao, MU, Tai-Jiang, ZHANG, Song-Hai, MARTIN, Ralph R, CHENG, Ming-Ming, HU, Shin-Min. **Attention Mechanisms in Computer Vision: A Survey**. Journal of Latex Vol. 14. Disponível em: <https://arxiv.org/pdf/2111.07624v1.pdf>. Acessado em: 12/10/2022.
  - [33] DAO, Son D, ZHAO, Ethan, PHUNG, Dinh, CAI, Jianfei. **Contrast Learning Visual Attention for Multi Label Classification**. Disponível em: <https://arxiv.org/pdf/2107.11626.pdf>. Acessado em: 07/10/2022.

- [34] VIDIT, Vidit, SALZMANN, Mathieu. **Attention-based domain adaptation for single-stage detectors.** Disponível em: <https://link.springer.com/article/10.1007/s00138-022-0132>. Acessado em: 24/10/2022.
- [35] XU, Bing, WANG, Naiyan, CHEN, Tianqi, LI, Mu. **Empirical Evaluation of Rectified Activations in Convolutional Network.** Disponível em: <https://arxiv.org/pdf/1505.00853>. Acessado em: 20/11/2022
- [36] MUNAWAR, Rizwan Muhammad. **How do Hyperparameters of YOLOv5 Work?.** Disponível em: <https://medium.com/augmented-startups/how-hyperparameters-of-yolov5-works-ec4d25f311a2>. Acessado em: 29/11/2022
- [37] VERCOSA, João Pedro Knauer de Q.. **Final-Thesis-Projects-Archives.** Disponível em: [https://drive.google.com/drive/folders/1-GTAjtc\\_tkahRe-0zcQx3Kfso2R\\_0Nq7?usp=sharing](https://drive.google.com/drive/folders/1-GTAjtc_tkahRe-0zcQx3Kfso2R_0Nq7?usp=sharing). Acessado em: 11/10/2022.
- [38] **Roboflow.** Disponível em: <https://roboflow.com/>. Acessado em: 12/09/2022
- [39] Free Software Foundation, Licença Pública Geral GNU (GPLv3). Disponível em: <https://www.gnu.org/licenses/gpl-3.0.pt-br.html> Acessado em: 07/12/2022
- [40] TAKEUCHI, Hirotaka, NONAKA, Ikujiro. **The New New Product Development Game.** Harvard Business Review, Janeiro, 1986. Disponível em: <https://web.archive.org/web/20210209210402/http://agilix.nl/resources/TheNewNewProductDevelopmentGame.pdf>. Acessado em: 07/12/2022.

## 9 – Apêndice

Tabela 1 – Lista, distribuição e descrição das classes das máscaras de placas de trânsito brasileiras

Classe	Código	Descrição	Número de amostras
Advertência	1a	Curva acentuada à esquerda	1
Advertência	1b	Curva acentuada à direita.	1
Advertência	2a	Curva à esquerda	1
Advertência	2b	Curva à direita	2
Advertência	3a	Pista sinuosa à esquerda	1
Advertência	3b	Pista sinuosa à direita	1
Advertência	4a	Curva acentuada em “S” à esquerda	1
Advertência	4b	Curva acentuada em “S” à direita	1
Advertência	5a	Curva em “S” à esquerda	1
Advertência	5b	Curva em “S” à direita	1
Advertência	6	Cruzamento de vias	1
Advertência	7a	Via lateral à esquerda	1
Advertência	7b	Via lateral à direita	1
Advertência	8	Interseção em “T”	1
Advertência	9	Bifurcação em “Y”	1
Advertência	10a	Entroncamento oblíquo à esquerda	1
Advertência	10b	Entroncamento oblíquo à direita	1
Advertência	11a	Junções sucessivas contrárias, primeira à esquerda	1



Advertência	11b	Junções sucessivas contrárias, primeira à direita	1
Advertência	12	Interseção em círculo	1
Advertência	13a	Confluência à esquerda	1
Advertência	13b	Confluência à direita	1
Advertência	14	Semáforo à frente	1
Advertência	15	Parada obrigatória à frente	1
Advertência	16	Bonde	1
Advertência	17	Pista irregular	1
Advertência	18	Saliência ou Lombada	1
Advertência	19	Depressão	1
Advertência	20a	Declive acentuado	1
Advertência	20b	Aclive acentuado	1
Advertência	21a	Estreitamento de pista ao centro	1
Advertência	21b	Estreitamento de pista à esquerda	1
Advertência	21c	Estreitamento de pista à direita	1
Advertência	21d	Alargamento de pista à esquerda	1
Advertência	21e	Alargamento de pista à direita	1
Advertência	22	Ponte estreita	1
Advertência	23	Ponte móvel	1
Advertência	24	Obras	1
Advertência	25	Mão dupla adiante	1
Advertência	26a	Sentido único	1
Advertência	26b	Sentido duplo	1
Advertência	27	Área com desmoronamento	1
Advertência	28	Pista escorregadia	1
Advertência	29	Projeção de cascalho	1
Advertência	30a	Trânsito de ciclistas	1
Advertência	30b	Passagem sinalizada de ciclistas	1
Advertência	30c	Trânsito compartilhado por ciclistas e pedestres	1
Advertência	31	Trânsito de tratores ou maquinaria agrícola	1
Advertência	32a	Trânsito de pedestres	1
Advertência	32b	Passagem sinalizada de pedestres	1
Advertência	33a	Área escolar	1
Advertência	33b	Passagem sinalizada de escolares	1
Advertência	34	Crianças	1
Advertência	35	Animais	1
Advertência	36	Animais selvagens	1
Advertência	37	Altura limitada	1
Advertência	38	Largura limitada	1
Advertência	39	Passagem de nível sem barreira	1
Advertência	40	Passagem de nível com barreira	1
Advertência	41	Cruz de Santo André	1
Advertência	42a	Início de pista dupla	1
Advertência	42b	Fim de pista dupla	1
Advertência	42c	Pista dividida	1
Advertência	43	Aeroporto	1
Advertência	44	Vento lateral	1
Advertência	45	Rua sem saída	1
Advertência	46	Peso bruto total limitado	1
Advertência	47	Peso limitado por eixo	1

Advertência	48	Comprimento limitado	1
Advertência	IC	Informações Complementares de Advertência	15
Advertência	SEA	Sinalização especial de Advertência	14
Indicação	E	Sinalização educativa	25
Indicação	I	Sinalização de identificação	16
Indicação	IK	Placa de identificação quilométrica	2
Indicação	O	Sinalização de orientação	17
Indicação	SA	Sinalização de serviços auxiliares	6
Indicação	SA01	Serviços Auxiliares: Pátios coletivos de estacionamento	1
Indicação	SA02	Serviços Auxiliares: Informações a turistas	1
Indicação	SA06	Serviços Auxiliares: Telefone para uso público	1
Indicação	SA07	Serviços Auxiliares: Atendimento mecânico	1
Indicação	SA08	Serviços Auxiliares: Serviço de borracheiro	1
Indicação	SA09	Serviços Auxiliares: Posto de abastecimento	1
Indicação	SA10	Serviços Auxiliares: Pronto-socorro, hospital ou casa de saúde	1
Indicação	SA11	Serviços Auxiliares: Sanitários públicos	1
Indicação	SA12	Serviços Auxiliares: Restaurante ou lanchonete	1
Indicação	SA13	Serviços Auxiliares: Serviços hoteleiros	1
Indicação	SA14	Serviços Auxiliares: Área destinada a camping	1
Indicação	SA18	Serviços Auxiliares: Praça de pedágio	1
Indicação	SA19	Serviços Auxiliares: Terminais de passageiros de ônibus urbano, interurbano ou rodoviário	1
Indicação	SA20	Serviços Auxiliares: Terminais e estações de trem e metrô	1
Indicação	SA21	Serviços Auxiliares: Aeroportos domésticos e internacionais	1
Indicação	SA24	Serviços Auxiliares: Balsa ou <i>ferry-boat</i>	1
Indicação	SA26	Serviços Auxiliares: Parada de ônibus	1
Regulamentação	1	Parada obrigatória	1
Regulamentação	2	Dê a preferência	2
Regulamentação	3	Sentido proibido	1
Regulamentação	4a	Proibido virar à esquerda	1
Regulamentação	4b	Proibido virar à direita	1
Regulamentação	5a	Proibido retornar à esquerda	1
Regulamentação	5b	Proibido retornar à direita	1
Regulamentação	6a	Proibido estacionar	1
Regulamentação	6b	Estacionamento regulamentado	1
Regulamentação	6c	Proibido parar e estacionar	1
Regulamentação	7	Proibido ultrapassar	1
Regulamentação	8a	Proibido mudar de faixa ou pista de trânsito da esquerda para direita	1

Regulamentação	8b	Proibido mudar de faixa ou pista de trânsito da esquerda para esquerda	1
Regulamentação	9	Proibido trânsito de caminhões	1
Regulamentação	10	Proibido trânsito de veículos automotores	1
Regulamentação	11	Proibido trânsito de veículos de tração animal	1
Regulamentação	12	Proibido trânsito de bicicletas	1
Regulamentação	13	Proibido trânsito de tratores e máquinas de obras	1
Regulamentação	14	Peso bruto total máximo permitido	1
Regulamentação	15	Altura máxima permitida	1
Regulamentação	16	Largura máxima permitida	1
Regulamentação	17	Peso máximo permitido por eixo	1
Regulamentação	18	Comprimento máximo permitido	1
Regulamentação	19	Velocidade máxima permitida	3
Regulamentação	20	Proibido acionar buzina ou sinal sonoro	1
Regulamentação	21	Alfândega	1
Regulamentação	22	Uso obrigatório de corrente	1
Regulamentação	23	Conserve-se à direita	1
Regulamentação	24a	Sentido de circulação da via ou pista	4
Regulamentação	24b	Passagem obrigatória	1
Regulamentação	25a	Vire à esquerda	1
Regulamentação	25b	Vire à direita	1
Regulamentação	25c	Siga em frente ou à esquerda	1
Regulamentação	25d	Siga em frente ou à direita	1
Regulamentação	26	Siga em frente	1
Regulamentação	27	Ônibus, caminhões e veículos de grande porte, mantenham-se à direita	1
Regulamentação	28	Duplo sentido de circulação	1
Regulamentação	29	Proibido trânsito de pedestres	1
Regulamentação	30	Pedestre, ande pela esquerda	1
Regulamentação	31	Pedestre, ande pela direita	1
Regulamentação	32	Circulação exclusiva de ônibus	1
Regulamentação	33	Sentido circular na rotatória	1
Regulamentação	34	Circulação exclusiva de bicicletas	1
Regulamentação	35a	Ciclista, transite à esquerda	1
Regulamentação	35b	Ciclista, transite à direita	1
Regulamentação	36a	Ciclistas à esquerda, pedestres à direita	1
Regulamentação	36b	Pedestres à esquerda, ciclistas à direita	1
Regulamentação	37	Proibido trânsito de motocicletas, motonetas e ciclomotores	1
Regulamentação	38	Proibido trânsito de ônibus	1
Regulamentação	39	Circulação exclusiva de caminhão	1
Regulamentação	40	Trânsito proibido a carros de mão	1
Regulamentação	IC	Sinalização de informações complementares	10
Turismo	SAU01	Serviços Auxiliares: Área de estacionamento	1
Turismo	SAU02	Serviços Auxiliares: Informações Turísticas	1
Turismo	SAU03	Serviços Auxiliares: Câmbio	1
Turismo	SAU04	Serviços Auxiliares: Correios	1
Turismo	SAU05	Serviços Auxiliares: Rua 24 horas	1

Turismo	SAU06	Serviços Auxiliares: Serviço telefônico	1
Turismo	SAU07	Serviços Auxiliares: Serviço Mecânico	1
Turismo	SAU08	Serviços Auxiliares: Abastecimento	1
Turismo	SAU09	Serviços Auxiliares: Borracharia	1
Turismo	SAU10	Serviços Auxiliares: Pronto socorro	1
Turismo	SAU11	Serviços Auxiliares: Serviço sanitário	1
Turismo	SAU12	Serviços Auxiliares: Restaurante	1
Turismo	SAU13	Serviços Auxiliares: Hotel	1
Turismo	SAU14	Serviços Auxiliares: Área de campismo	1
Turismo	SAU15	Serviços Auxiliares: Estacionamento de trailer	1
Turismo	SAU19	Serviços Auxiliares: Terminal rodoviário	1
Turismo	SAU20	Serviços Auxiliares: Terminal ferroviário	1
Turismo	SAU21	Serviços Auxiliares: Aeroporto	1
Turismo	SAU22	Serviços Auxiliares: Heliporto	1
Turismo	SAU23	Serviços Auxiliares: Porto	1
Turismo	SAU24	Serviços Auxiliares: Transporte sobre a água	1
Turismo	SAU25	Serviços Auxiliares:	1
Turismo	SAU26	Serviços Auxiliares: Ponto de parada	1
Turismo	TAD01	Área para prática de esportes: Esportes	1
Turismo	TAD02	Área para prática de esportes: Esportes equestres	1
Turismo	TAD03	Área para prática de esportes: Esportes automobilísticos	1
Turismo	TAD04	Área para prática de esportes: Esportes náuticos	1
Turismo	TAD05	Área para prática de esportes: Mergulho	1
Turismo	TAD06	Área para prática de esportes: Voo Livre	1
Turismo	TAD07	Área para prática de esportes: Surfe	1
Turismo	TAD08	Área para prática de esportes: Canoagem	1
Turismo	TAD09	Área para prática de esportes: Pesca submarina	1
Turismo	TAD10	Área para prática de esportes: Pesca esportiva	1
Turismo	TAD11	Área para prática de esportes: Montanhismo	1
Turismo	TAD12	Área para prática de esportes: Golfe	1
Turismo	TAD13	Área para prática de esportes: Aeroclube	1
Turismo	TAD14	Área para prática de esportes: Marina	1
Turismo	TAD15	Área para prática de esportes: Futebol	1
Turismo	TAD16	Área para prática de esportes: Ciclismo	1
Turismo	TAR01	Áreas de recreação: Praça	1
Turismo	TAR02	Áreas de recreação: Barco de passeio	1
Turismo	TAR03	Áreas de recreação: Parque urbano	1
Turismo	TAR04	Áreas de recreação: Represa	1
Turismo	TAR05	Áreas de recreação: Teleférico	1
Turismo	TAR06	Áreas de recreação: Mirante	1
Turismo	TAR07	Áreas de recreação: Parque de diversões	1
Turismo	THC01	Atrativos históricos e culturais: Arquitetura religiosa	1

Turismo	THC02	Atrativos históricos e culturais: Arquitetura militar	1
Turismo	THC03	Atrativos históricos e culturais: Arquitetura histórica	1
Turismo	THC04	Atrativos históricos e culturais: Monumento	1
Turismo	THC05	Atrativos históricos e culturais: Museu	1
Turismo	THC06	Atrativos históricos e culturais: Ruína	1
Turismo	THC07	Atrativos históricos e culturais: Patrimônio cultural	1
Turismo	THC08	Atrativos históricos e culturais: Sítio arqueológico	1
Turismo	THC09	Atrativos históricos e culturais: Farol	1
Turismo	THC10	Atrativos históricos e culturais: Centro de cultura	1
Turismo	THC11	Atrativos históricos e culturais: Biblioteca	1
	TIT01	Locais de interesse turístico: Festas Populares	1
Turismo	TIT02	Locais de interesse turístico: Teatro	1
Turismo	TIT03	Locais de interesse turístico: Convenções	1
Turismo	TIT04	Locais de interesse turístico: Artesanato	1
Turismo	TIT05	Locais de interesse turístico: Zoológico	1
Turismo	TIT06	Locais de interesse turístico: Planetário	1
Turismo	TIT07	Locais de interesse turístico: Feira típica	1
Turismo	TIT08	Locais de interesse turístico: Exposição Agropecuária	1
Turismo	TIT09	Locais de interesse turístico:	1
Turismo	TIT10	Locais de interesse turístico: Pavilhão de exposições	1
Turismo	TNA01	Atrativos turísticos naturais: Montanha	1
Turismo	TNA02	Atrativos turísticos naturais: Praia	1
Turismo	TNA03	Atrativos turísticos naturais: Ilha	1
Turismo	TNA04	Atrativos turísticos naturais: Lagoa	1
Turismo	TNA05	Atrativos turísticos naturais: Cachoeira	1
Turismo	TNA06	Atrativos turísticos naturais: Patrimônio cultural	1
Turismo	TNA07	Atrativos turísticos naturais: Gruta	1
Turismo	TNA08	Atrativos turísticos naturais: Turismo rural	1
Turismo	TNA09	Atrativos turísticos naturais: Estância hidromineral	1
Turismo	I	Sinalização turística de indicação	11
Outros	FE	Fiscalização eletrônica	14
Outros	MA	Marcador de alinhamento	1
Outros	MP	Marcador de perigo	8
Outros	O	Sinalização geral de obras	5
Outros	OA15	Sinalização de obras similar a A-15	1
Outros	OA17	Sinalização de obras similar a A-17	1
Outros	OA18	Sinalização de obras similar a A-18	1
Outros	OA19	Sinalização de obras similar a A-19	1
Outros	OA21a	Sinalização de obras similar a A-21a	1
Outros	OA21b	Sinalização de obras similar a A-21b	1

Outros	OA21c	Sinalização de obras similar a A-21c	1
Outros	OA24	Sinalização de obras similar a A-24	1
Outros	OA25	Sinalização de obras similar a A-25	1
Outros	OA27	Sinalização de obras similar a A-27	1
Outros	OA28	Sinalização de obras similar a A-28	1
Outros	OA29	Sinalização de obras similar a A-29	1
Outros	OA37	Sinalização de obras similar a A-37	1
Outros	OA38	Sinalização de obras similar a A-38	1
Outros	OA42	Sinalização de obras similar a A-42	1
Outros	SC	Sinalização de Curva Acentuada	3

**Tabela 2 – Distribuição de classes: simpleDataset**

Código	Número de amostras
'R-19'	230
'O-MP'	212
'R-24a'	141
'R-2'	115
'R-6c'	92
'R-15'	90
'A-32b'	83
'A-24'	82
'A-32a'	77
'R-5a'	76
'R-3', 'R-5b'	74
'I-E'	71
'I-SA'	60

**Tabela 3 – Distribuição de classes: normalDataset**

Código	Número de amostras
'R-19'	196
'O-MP'	180
'R-24a'	107
'I-SA'	96
'R-2'	81
'R-6c'	58
'R-15'	56
'A-32b'	49
'A-24'	48
'A-32a', 'R-5a'	42
'R-3', 'R-5b', 'R-25b', 'T-SAU24', 'O-OA37', 'R-6b', 'A-31', 'R-39', 'A-27', 'R-28', 'R-4a', 'T-TAD06', 'A-14', 'R-10', 'T-TAD11', 'R-22', 'A-13a', 'R-34', 'A-33a', 'R-9', 'T-THC05', 'O-OA42a', 'T-SAU06', 'T-TIT06', 'R-21', 'R-18', 'A-4b', 'A-40', 'R-25d', 'T-TIT04'	39
'A-48', 'A-37', 'T-SAU08', 'T-TIT08', 'A-30b', 'I-SA18', 'A-10b', 'O-OA38', 'O-OA18', 'A-34', 'R-6a', 'T-SAU13', 'O-OA19', 'T-THC04', 'R-40', 'T-TAR02', 'A-11a', 'T-TAR06', 'I-SA12', 'R-30', 'R-32', 'R-16', 'I-SA20', 'A-22', 'T-SAU07', 'A-29', 'R-17', 'A-38', 'O-	38

OA21b', 'T-TAD15', 'T-SAU20', 'T-TAR05', 'I-SA02', 'R-33', 'R-25a', 'I-SA24', 'R-8b', 'A-25', 'A-16', 'T- TIT09', 'T-TNA03', 'T-TAD04', 'O-OA28', 'R-1', 'A-8', 'T-TIT07', 'R-26', 'R-36b', 'T-THC07', 'T-THC09', 'T- SAU01', 'T-SAU25', 'A-42a', 'A-7a', 'R-37', 'A-46', 'T- THC10', 'R-13', 'T-SAU11', 'T-TIT01', 'A-28', 'A-19', 'R-14', 'R-36a', 'T-SAU12', 'T-TIT05', 'A-', 'I-SA07', 'T- TNA02', 'R-4b', 'O-OA29', 'I-IK', 'A-6', 'I-SA01', 'O- OA17', 'T-TAR04', 'A-23', 'T-SAU04', 'R-24b', 'R-20', 'A-30a', 'A-4a', 'T-THC01', 'T-TAD05', 'T-TNA01', 'R- 25b', 'T-THC11', 'O-OA15', 'A-10a', 'T-SAU19', 'T- TNA05', 'T-SAU22', 'R-11', 'T-TIT03', 'A-2a', 'I-SA14', 'T-TAD14', 'A-20b', 'T-TIT10', 'A-36', 'A-18', 'T- SAU09', 'I-SA08', 'T-SAU10', 'T-TNA08', 'T-TAR03', 'O-OA27', 'A-15', 'A-11b', 'T-TIT02', 'A-9', 'T-TNA07', 'T-TAD08', 'I-SA26', 'I-SA06', 'T-TAD12', 'A-20a', 'R- 8a', 'A-1a', 'A-44', 'T-TNA04', 'A-21e', 'I-SA10', 'A-45', 'T-SAU03', 'A-17', 'T-TNA06', 'O-MA', 'R-38', 'A-21d', 'T-THC08', 'A-21a', 'A-42c', 'T-SAU02', 'T-TAD02', 'A-1b', 'T-TAD07', 'O-SC', 'T-TAR07', 'T-SAU14', 'T- TAD16', 'R-23', 'A-3b', 'T-TAD10', 'I-SA21', 'R-29', 'A- 33b', 'T-SAU23', 'T-TAD01', 'R-IC', 'T-TAD03', 'I- SA13', 'A-3a', 'A-7b', 'R-27', 'R-35b', 'A-13b', 'R-7', 'T- TAD13', 'T-SAU15', 'I-O', 'A-SEA', 'T-TAD09', 'A- 42b', 'R-12', 'A-21b', 'I-SA09', 'A-41', 'A-12', 'T- SAU05', 'T-SAU26', 'T-TNA09', 'T-THC02', 'R-35a', 'T-THC03', 'A-5a', 'O-OA21a', 'T-THC06', 'T-SAU21', 'R-25c', 'O-OA25', 'I-SA19', 'A-30c', 'R-31', 'A-5b', 'A- 2b'	
'A-43', 'O-FE', 'O-O', 'A-26a', 'T-I', 'I-E', 'A-47', 'O- OA21c'	37
'I-I', 'A-35', 'A-26b', 'I-SA11', 'A-IC'21c',	36
'O-OA24'	29