



UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica e de Computação

ALYSSON GOMES DE OLIVEIRA

**MODELO DE INTELIGÊNCIA
ARTIFICIAL COM VISÃO COMPUTACIONAL APLICADO EM CARROS
AUTÔNOMOS USANDO SIMULADOR**

CAMPINAS

2021

ALYSSON GOMES DE OLIVEIRA

**MODELO DE INTELIGÊNCIA
ARTIFICIAL COM VISÃO COMPUTACIONAL APLICADO EM CARROS
AUTÔNOMOS USANDO SIMULADOR**

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na área de Telecomunicações e Telemática.

Supervisor/Orientador: Prof. Dr. Yuzo Iano

Este trabalho corresponde à versão final da dissertação defendida pelo aluno Alysson Gomes de Oliveira, orientada pelo(a) Prof. Dr. Prof. Dr. Yuzo Iano

Assinatura do Orientador

CAMPINAS

2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

OL4m Oliveira, Alysson Gomes de, 1980-
Modelo de inteligência artificial com visão computacional aplicado em carros autônomos usando simulador / Alysson Gomes de Oliveira. – Campinas, SP : [s.n.], 2021.

Orientador: Yuzo Iano.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Veículos autônomos. 2. Aprendizado profundo. 3. Inteligência artificial. 4. Aprendizado por reforço. 5. Visão computacional. 6. Aprendizado de máquina. I. Iano, Yuzo, 1950-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Artificial intelligence model with computational vision applied to autonomous cars using simulator

Palavras-chave em inglês:

Autonomous vehicles

Deep learning

Artificial Intelligence

Reinforcement learning

Computer vision

Machine learning

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre em Engenharia Elétrica

Banca examinadora:

Yuzo Iano [Orientador]

Diogo Gará Caetano

Rosivaldo Ferrarezi

Data de defesa: 19-05-2021

Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-8813-0074>

- Currículo Lattes do autor: <http://lattes.cnpq.br/2245796893444865>

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Alysson Gomes de Oliveira RA: 226530

Data da Defesa: 19 de maio de 2021

Título da Tese: "Modelo de inteligência artificial com visão computacional aplicado em carros autônomos usando simulador".

Prof. Dr. Yuzo Iano (Presidente)

Prof. Dr. Diogo Gará Caetano

Prof. Dr. Rosivaldo Ferrarezi

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

AGRADECIMENTOS

Só foi possível chegar à conclusão deste trabalho devido ao apoio de várias pessoas. Em primeiro lugar, não posso deixar de agradecer ao meu orientador, Professor Doutor Yuzo Iano, pela paciência, empenho e sentido prático com que sempre me orientou, além das lições de vidas que jamais serão esquecidas. Muito obrigado por ter corrigido, quando necessário, sem nunca me desmotivar. Posso garantir que sem essa ajuda, nada disso seria possível.

Estendo o agradecimento a todos os estudantes e professores da FEEC e aos colaboradores e colegas do LCV, guardo muitas boas lembranças. Finalmente, quero agradecer à minha família e amigos pelo apoio incondicional que me deram, especialmente a minha esposa pelas revisões incansáveis ao longo da elaboração deste trabalho.

A todos, deixo o meu mais sincero agradecimento.

“A educação é a arma mais poderosa que você pode usar para mudar o mundo.”

Nelson Mandela

RESUMO

O trabalho de pesquisa desenvolvido nesta dissertação, evidência pontos que precisam ser levados em consideração ao pensarmos no desenvolvimento de carros autônomos.

A forma de locomoção no futuro será afetada pelas novas tecnologias, por isso a importância de compreender e diminuir o impacto causado pela transição do método tradicional ao método completamente automatizado.

O foco principal deste estudo é desenvolver modelos de inteligência artificial que forneçam melhor performance em condições extremas do meio. A metodologia baseou-se na exploração científico-comparativa de diferentes modelos e, a partir dos resultados obtidos, chegar a uma conclusão a respeito do melhor modelo a ser adotado.

Ao longo do trabalho serão apresentados os embasamentos de cada modelo criado e as suas devidas aplicações para o desenvolvimento do melhor modelo de inteligência artificial.

Palavras-chave: Inteligência artificial, carro autônomo, visão computacional, *machine learning*, *deep learning*.

ABSTRACT

This research work that was developed in this dissertation, shows how some points need to be considered when we think about the development of autonomous cars. We know that the way we move around in the future will be affected by these new technologies, and so we must think about how we can understand and reduce the impact of the transition from the traditional method to the fully automated method. The main objective of this work is to show how we can develop artificial intelligence models to have a better performance in extreme environment conditions. The methodology that was used in this work is the scientific exploration of comparing several models with their respective results to finally reach a conclusion of the best model to be applied. Throughout the work, the basis of each mathematical model and its proper application for the development of the best artificial intelligence model will be presented.

Key Words: Artificial intelligence, autonomous car, computer vision, machine learning, deep learning.

LISTA DE ILUSTRAÇÕES

Figura 1.1.1 - Comparação das cidades de Boston, Berlin e Shangai da mudança do novo modal de transporte urbano.	19
Figura 1.1.2 - Porcentagem de pessoas que concordam que carros autônomos não serão seguros.	20
Figura 1.1.3 - Investigadores do Departamento Nacional de segurança de Transporte (NTSB) examinam um Carro Autônomo da Uber que ocasionou um atropelamento fatal em Tempe, Arizona em 20 de março de 2018.....	20
Figura 2.1.1 - Pavilhão da General Motors na Feira Mundial 1939 em Nova York	24
Figura 2.1.2 - Pavilhão da Highways and Horizons.....	24
Figura 2.1.3 - Um 1926 Chandler controlado por rádio controle em 1925 em NY.....	25
Figura 2.1.4 - O Carro da Universidade Stanford concluindo o percurso no DARPA 2005	26
Figura 2.1.5 - Classificação dos carros autônomos	26
Figura 2.1.1 - Smart Roads pode se comunicar entre as rodovias e usuários igual a um objeto conectado na rede da internet com atualizações de informações	29
Figura 3.1.1 - Característica do funcionamento do aparelho piezoelétrico	31
Figura 3.2.1 - Rodovia Musical de Lancaster na Califórnia	33
Figura 3.3.1 - Instalação de linhas de carregamento para veículos elétricos do projeto eRoadArlanda na Suécia. O governo tem um planejamento de expansão nacional do projeto ...	35
Figura 3.3.2 - Rodovias Elétricas que carregam veículos elétricos enquanto estão na linha verde de carregamento, sem precisar parar o percurso.....	35
Figura 3.4.1 - Conceito do Poste programável de sinais de trânsito	36
Figura 3.5.1 - Tipos de comunicação dentro da arquitetura V2X	37
Figura 3.6.1 - Em (a) os possíveis acidentes que podem ocorrer na intersecção e em (b) o ponto cego do motorista em uma intersecção.....	40
Figura 3.6.2 - Exemplo de uma intersecção inteligente que usa câmeras com reconhecimento de objetos e comunicação V2X para avisar SDC antecipadamente	40
Figura 3.6.1 - Conversão da imagem digital.....	42
Figura 3.6.2 - Decomposição de imagem colorida em matrix RGB	43
Figura 3.6.3 - Representação matricial de imagem colorida	44
Figura 4.1.1 - Imagem original sem tratamento do método cvtColor().....	45
Figura 4.1.2 - Diagrama da função cvtColor()	46
Figura 4.1.3 - Resultado da imagem após o uso do método cvtColor().....	46
Figura 4.1.4 - Diagrama da função de Blur Gaussiano do OpenCV	47
Figura 4.1.5 - Imagem original à esquerda e a imagem à direita com a suavização do filtro Gaussiano	47
Figura 4.1.6 - Imagem original sem aplicação de filtro	49
Figura 4.1.7 - Imagem pré-processada usando filtros de conversão de cores para cores de cinza	49
Figura 4.1.8 - Histograma da imagem após a aplicação do filtro de conversão de cores para cores de cinza.....	50
Figura 4.1.9 - Diagrama da função de binarização de uma imagem.....	50
Figura 4.1.10 - Imagem após aplicação da binarização da mesma.....	50
Figura 4.1.11 - Histograma da imagem com o filtro da binarização aplicado	51
Figura 4.2.1 - Definições de Haars	51
Figura 4.4.1 - Subdivisões da Inteligência Artificial.....	55
Figura 5.2.1 - Estrutura matemática de funcionamento de um <i>neuron</i>	57

Figura 5.2.2 - Modelo simplificado do funcionamento de uma CNN	58
Figura 5.3.1 - Métodos de aprendizagem no <i>Deep Learning</i>	60
Figura 5.4.1 - Loop de funcionamento do Modelo	63
Figura 5.4.2 - Trajetória dividida em Grid e com pontos atribuídos.....	64
Figura 5.4.3 -Interação acumulando pontos.....	64
Figura 6.1.1 - Diagrama da criação de <i>dataset</i>	66
Figura 6.1.2 - Configuração do SDC com uma câmera frontal.....	67
Figura 7.1.1 - Parâmetro de posicionamento na trajetória	68
Figura 7.2.1 - Parâmetro de <i>Heading</i> do SDC	69
Figura 7.3.1 - Parâmetro de distância do SDC até a linha central da trajetória	70
Figura 7.4.1 - Parâmetro que verifica se todas as rodas estão dentro da trajetória.....	71
Figura 7.5.1 - Parâmetro de velocidade do SDC	72
Figura 7.6.1 - Parâmetro do ângulo da direção do SDC.....	73
Figura 7.7.1 - Diagrama de um exemplo da função recompensa	73
Figura 7.7.2 - Visão da simulação do SDC no AWS	74
Figura 7.7.3 - Diagrama da versão do modelo RL	75
Figura 8.1.1 - Modelo e descrição da trajetória usada no desenvolvimento das versões do modelo RL.....	78
Figura 8.5.1 - Track com um grau maior de dificuldade para ser usado no modelo RL V5	85
Figura 8.5.2 - Fluxograma lógico do acesso ao algoritmo no modo de segurança do modelo RL.....	86
Figura 8.5.3 - ScreenShot da simulação do modelo RL V5 no segundo track no simulador da AWS	87
Figura 8.5.4 - Modelo simplificado de um Carro Autônomo	88
Figura 8.5.5 - Fluxo do modelo do Carro Autônomo	88
Figura 9.1.1 - Pesquisa da KPMG sobre política e legislação em relação a mobilidade autônoma	90
Figura 9.1.2 - Pesquisa da KPMG sobre tecnologia e inovação em relação a mobilidade autônoma	91
Figura 9.1.3 - Pesquisa da KPMG sobre infraestrutura em relação a mobilidade autônoma	91
Figura 10.1.1 - Propaganda em 1956 de como seriam os carros autônomos do futuro	93
Figura 10.1.2 - Projeto do Google de Carros Autônomos que se iniciou em 2009.....	94
Figura 10.1.3 - Projeto do Google Waymo Via na área de logística	95
Figura 10.1.4 - Projeto piloto da Empresa Uber com carros autônomos	95
Figura 10.1.5 - Projeto da Pizzaria Domino's de entrega usando carros autônomos em parceria com a startup Muro	96
Figura 10.1.6 - Veículo autônomo da startup Nuro sendo carregado para entrega de pizzas da Domino's	96
Figura 10.1.7 - Visão de como vai ser o transporte em Londres em 2045	97
Figura 13.1.1 - Interface inicial do Google Colab.....	103
Figura 13.2.1 - Carro autônomo da AWS visão lateral.....	104
Figura 13.2.2 - Visão frontal do carro autônomo da AWS.....	105
Figura 13.2.3 - Visão lateral do carro autônomo da AWS.....	105
Figura 13.2.4 - Modelo do AWS <i>DeepRacer</i> Evo, que tem câmera estéreo e um sensor Lidar...106	106
Figura 13.2.5 - Visão lateral do SDC da AWS com sensor Lidar	107
Figura 13.2.6 - Visão em perspectiva do AWS <i>DeepRacer</i> Evo com câmeras Estéreo e sensor Lidar.....	107
Figura 13.2.7 - Inclusão de mais uma câmera frontal e um sensor Lidar	108
Figura 13.2.8 - Inclusão do Sensor Lidar no AWS <i>DeepRacer</i>	108

Figura 13.2.9 - Interface Inicial da AWS DeepRacer	109
Figura 13.2.10 - Amazon SageMaker Interface para ML	109
Figura 13.2.11 - Amazon SageMaker Tela Inicial	110
Figura 14.1.1 - - ScreenShot da Evaluation do 4º do Modelo	111
Figura 14.1.2 - ScreenShot da Evaluation da 5ª versão do modelo em uma outra pista	111
Figura 15.1.1 - Carro autônomos fica esperando na área de abastecimento da pizzeria	112
Figura 15.1.2 - Funcionário acomoda a pizza no compartimento de produto do carro autônomo	113
Figura 15.1.3 - Cliente recebe uma notificação que seu pedido já foi carregado e já saiu para a fazer a entrega	113
Figura 15.1.4 - A Empresa consegue acompanhar o trajeto do carro autônomo em tempo real	114
Figura 15.1.5 - Ao chegar o SDC, é enviado uma notificação ao cliente sobre a chegada do produto	114
Figura 15.1.6 - Cliente entra com seu código único do seu pedido para a retirada do seu produto	115
Figura 15.1.7 - Após o cliente entrar com seu código, o SDC libera para o cliente a retirada do seu produto	116
Figura 16.1.1 - Página principal do Simulador Carla	117
Figura 16.1.2 - Empresas envolvidas no desenvolvimento do simulador Carla	117
Figura 16.2.1 - Screen Shot do Simulador da Udacity.....	118
Figura 16.3.1 - Simulador da Nvidia	118
Figura 16.4.1 - AirSim -simulador open source da Microsoft que pode ser usado até para simulação de drones	119
Figura 17.1.1 - Exemplo da aplicação do método cvtColor()	120
Figura 17.1.2 - Exemplo de aplicação do filtro Gaussiano de suavização.....	120
Figura 17.1.3 - Aplicação da função threshold na imagem em gradientes de cinza para obter uma imagem binária.....	121
Figura 17.1.4 - Trecho do código para obtenção do dataset.....	121
Figura 17.1.5 - Desenvolvimento da Função de recompensa.....	121
Figura 17.1.6 - Versão da Função de recompensa do modelo RL.....	122

LISTA DE TABELAS

Tabela 3.1.1 - Resumo dos 10 avanços mais significativos nas rodovias inteligentes.....	30
Tabela 3.1.2 - Desenvolvimento e aplicação de rodovias piezoelétricas no mundo.....	31
Tabela 3.1.3 -Desenvolvimento de rodovias solares na China, França e Estados Unidos	32
Tabela 3.2.1 - Implantação do uso das rodovias músicas ao redor do mundo	33
Tabela 3.5.1 - Diferentes tipos de comunicação veicular e suas características	38
Tabela 3.6.1 - Exemplos de aplicação de intersecções inteligentes ao redor do mundo	41
Tabela 8.1.1 - Hyperparâmetros do CNN para as versões.....	77
Tabela 8.1.2 - Evolução da Versão 1 do Modelo RL.....	77
Tabela 8.2.1 - Evolução da versão 2 do modelo RL	79
Tabela 8.3.1 - Evolução da versão 3 do modelo RL	81
Tabela 8.4.1 - Evolução da versão 4 do modelo RL	82
Tabela 8.5.1 - Evolução da versão 5 do modelo RL	84
Tabela 8.5.2 - Resultados comparativos das versões dos modelos RL.....	85
Tabela 12.2.1 - Configuração do carro autônomo da AWS	106
Tabela 12.2.2 - Configuração do carro autônomo da AWS	106
Tabela 12.2.3 - Configurações do AWS DeepRacer EVO.....	109

LISTA DE GRÁFICOS

Gráfico 8.1.1 - Gráfico do desempenho da versão 1 do modelo RL	76
Gráfico 8.2.1 - Gráfico da Versão 2 do modelo RL.....	79
Gráfico 8.3.1 - Gráfico da versão 3 do modelo RL	80
Gráfico 8.4.1 - Gráfico da versão 4 do modelo RL	82
Gráfico 8.5.1 - Gráfico da versão 5 do modelo RL	84

LISTA DE EQUAÇÕES

Equação 4.1.1 - Operação matricial de conversão de cores.....	45
Equação 4.3.1 - Fórmula para detecção de lado usando Canny.....	52
Equação 4.4.1 - Representação da eq. da reta no sistema polar	54
Equação 5.2.1 - Equação de um neurônio artificial	57
Equação 5.3.1 - Equação matemática do método de aprendizagem por reforço baseado no valor	62
Equação 5.3.2 - Equação estocástica do modelo da política-baseado	62

LISTA DE ABREVIATURAS E SIGLAS

AWS	<i>Amazon Web Services</i>
CNN	<i>Convolutional Neural Network</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
DL	<i>Deep Learning</i>
IA	Inteligência Artificial
IOT	<i>Internet of Things</i>
PPO	<i>Proximal Policy Optimization Algorithms</i>
RL	<i>Reinforcement Learning</i>
NHTSA	<i>National Highway Traffic Safety Administration</i>
SDC	<i>Self Drive Car</i>
SR	<i>Smart Road</i>
V2I	<i>Vehicle-to-Infrastructure</i>
V2N	<i>Vehicle-to-Network</i>
V2P	<i>Vehicle-to-Pedestrian</i>
V2V	<i>Vehicle-to-Vehicle</i>

SUMÁRIO

1	INTRODUÇÃO	18
1.1	MOTIVAÇÃO	21
1.2	PROBLEMA A SER RESOLVIDO	21
1.3	OBJETIVOS	22
2	A HISTÓRIA DOS CARROS AUTÔNOMOS.....	22
2.1	DESENVOLVIMENTO TEÓRICO	23
3	SMART ROADS.....	28
3.1	RODOVIAS QUE PRODUZEM ENERGIA	30
3.2	RODOVIAS QUE PRODUZEM MÚSICAS	32
3.3	RODOVIAS QUE CARREGAM AUTOMATICAMENTE VEÍCULOS	34
3.4	RODOVIAS COM SINAIS DE TRÂNSITO SEM FIO INTELIGENTE	36
3.5	RODOVIAS COM V2X E VANETS.....	37
3.6	RODOVIAS COM INTERSECÇÕES INTELIGENTES.....	38
4	VISÃO COMPUTACIONAL	41
4.1	FRAMEWORKS: OPENCV	44
4.2	CLASSIFICADOR HAAR CASCADE	51
4.3	CANNY DETECÇÃO DE LADOS.....	52
4.4	TRANSFORMADA DE LINHA DE HOUGH.....	53
5	EM BUSCA DA INTELIGÊNCIA ARTIFICIAL	54
5.1	DEEP LEARNING.....	56
5.2	CONVOLUTIONAL NEURAL NETWORKS	56
5.3	APRENDIZAGEM POR REFORÇO (RL).....	60
5.4	APLICAÇÃO DA APRENDIZAGEM POR REFORÇO NO PROJETO	63
6	METODOLOGIA DE PESQUISA	65
6.1	AS FERRAMENTAS	65
7	DESENVOLVIMENTO DO ALGORITMO DO MODELO.....	66
7.1	PARÂMETRO DE POSICIONAMENTO NA TRAJETÓRIA.....	67
7.2	PARÂMETRO DE HEADING DO SDC	69
7.3	PARÂMETRO DA DISTÂNCIA AO CENTRO DA LINHA	69
7.4	PARÂMETRO DE TODAS AS RODAS NA TRAJETÓRIA.....	70
7.5	PARÂMETRO DE VELOCIDADE.....	71

7.6	PARÂMETRO DO ÂNGULO DA DIREÇÃO	72
7.7	CODIFICAÇÃO DO ALGORITMO COM OS PARÂMETROS	73
8	ANÁLISE DE DADOS	75
8.1	VERSÃO 1 DO MODELO RL	76
8.2	VERSÃO 2 DO MODELO RL	78
8.3	VERSÃO 3 DO MODELO RL	80
8.4	VERSÃO 4 DO MODELO RL	81
8.5	VERSÃO 5 DO MODELO RL	83
9	CARROS AUTÔNOMOS E O CENÁRIO BRASILEIRO	89
9.1	ÍNDICE DE PRONTIDÃO PARA VEÍCULOS AUTÔNOMOS 2020.....	89
10	CONCLUSÕES.....	92
10.1	UM PASSO PARA O FUTURO	92
11	TRABALHOS FUTUROS.....	98
12	REFERÊNCIAS.....	99
13	ANEXO I – FERRAMENTAS PARA O DESENVOLVIMENTO DO MODELO DL	103
13.1	GOOGLE COLAB	103
13.2	AWS DEEP RACER	104
14	ANEXO II – FOTOS DA SIMULAÇÃO DOS MODELOS NA AWS DEEPRACER.....	111
14.1	TESTANDO O MODELO	111
15	ANEXO III – SEQUÊNCIA DE FUNCIONAMENTO DA ENTREGA DE PIZZA POR CARRO AUTÔNOMO	112
15.1	DELIVERY AUTÔNOMO.....	112
16	ANEXO IV – SIMULADORES DE CARROS AUTÔNOMOS.....	117
16.1	CARLA	117
16.2	UDACITY SELF-DRIVE CAR.....	118
16.3	NVIDIA DRIVE SIM	118
16.4	AIR SIM	119
17	APENDICE A – CÓDIGOS	120
17.1	CODIGOS PARA PROCESSAMENTO DE IMAGEM.....	120

1 INTRODUÇÃO

“Como podemos fazer com que os carros se tornem autônomos?” Essa seria uma das consequências do avanço da robótica. No Seriado *Knight Rider*, o carro apresentado como *KITT* ou *Knight Industries Two Thousand* (LARSON), é um exemplo de inteligência artificial aplicado em um carro. Mas desde os primórdios dos estudos da robótica, a humanidade ansiou por uma certa mobilidade controlada por IA (Inteligência Artificial).

Com o avanço de várias frentes da tecnologia, pode-se dizer que o sonho de se ter um carro autônomo está cada vez mais próximo. Segundo um estudo realizado pelo *world economic forum* (MOAVENZADEH e LANG, 2018), foi feita uma previsão de como será impactada a mobilidade urbana em Boston com carros autônomos (Fig 1.1.1). Nesse gráfico é possível notar que o modal de transporte já considera SDC (*SELF DRIVE CAR*) a forma predominante em um futuro não distante. Os benefícios dos carros autônomos são inúmeros. Dentre eles destaca-se a segurança como fator determinante, 94% dos acidentes são ocasionados por falha humana (UNITED STATES DEPARTMENT OF TRANSPORTATION, 2019) segundo o Departamento de Rodovia Americano (NHTSA). Com o desenvolvimento do SDC essa estatística pode ser alterada.

Apesar de todos os benefícios que o SDC traz, a opinião pública ainda está em dúvida em relação à segurança dos carros autônomos (Fig 1.1.1 e Fig. 1.1.2). Um estudo de 2020 revela que, a população de países em desenvolvimento mostra maior apreensão quanto ao uso de carros autônomos em comparação à de países mais prósperos economicamente (2020 Global Automotive Consumer Study Is consumer interest in advanced automotive technologies on the move? 2020).

Em março de 2018, houve o primeiro acidente fatal envolvendo um carro autônomo e um pedestre (Fig 1.1.3). O relatório que a NTSB apresentou mostra que o SDC Uber conseguiu identificar o pedestre a 5,6s antes do impacto, mas não reconheceu corretamente como uma pessoa e, assim, ativar o modo correto para desviar ou parar o SDC. Apesar desse triste incidente, concluiu-se que uma corrente de fatores podem ter proporcionado inclusive a desatenção do motorista.

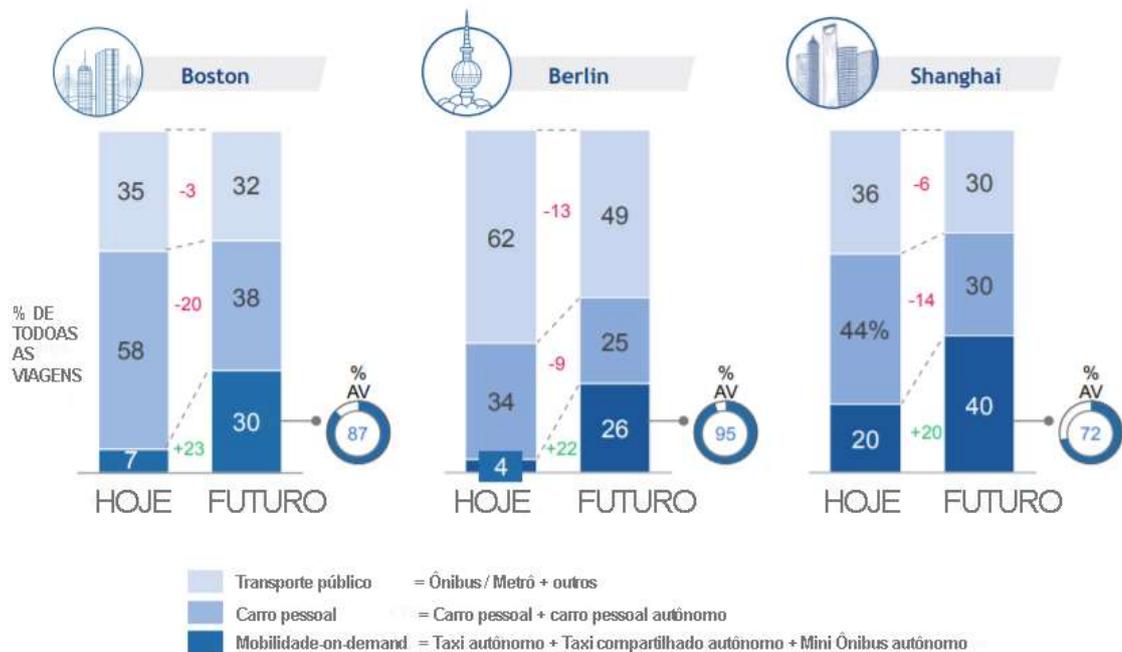
Os veículos autônomos precisarão não apenas obedecer às leis de trânsito, mas também se adaptar às condições da estrada e implementar estratégias de direção que levem em conta a incerteza sobre o que os outros usuários da estrada

farão, como dirigir mais devagar do que um motorista humano faria em áreas com alto número de pedestres ou em condições de baixa visibilidade.

Neste trabalho vamos desenvolver um algoritmo para condições extremas, nas quais o carro autônomo não poderá contar com o auxílio de grande parte dos seus sensores durante a navegação. Neste intuito, vamos empregar vários conceitos para ampliar a inteligência artificial de nosso protótipo de carro autônomo.

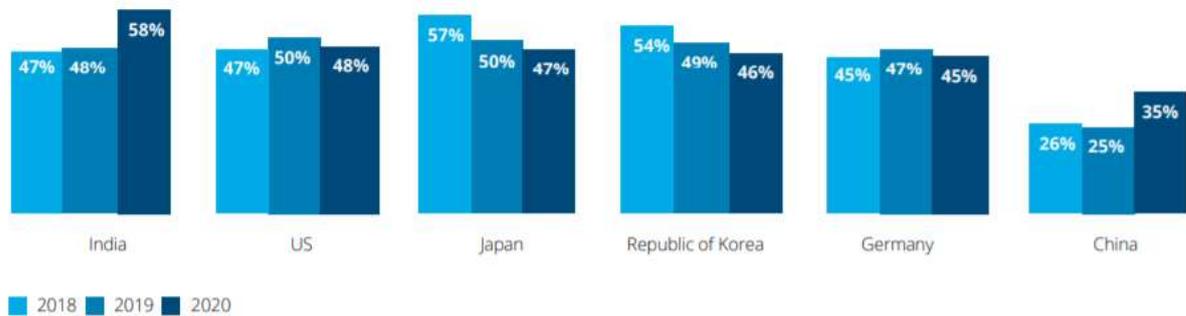
Um questionamento primordial do nosso estudo refere-se a como o carro vai conseguir identificar ruas, carros, objetos e pessoas para poder se locomover em condições extremas, privados dos muitos recursos de mobilidade. Nos serviremos do conceito de visão computacional e *deep learning* para que o carro se guie autonomamente.

Figura 1.1.1 - Comparação das cidades de Boston, Berlin e Shanghai da mudança do novo modal de transporte urbano.



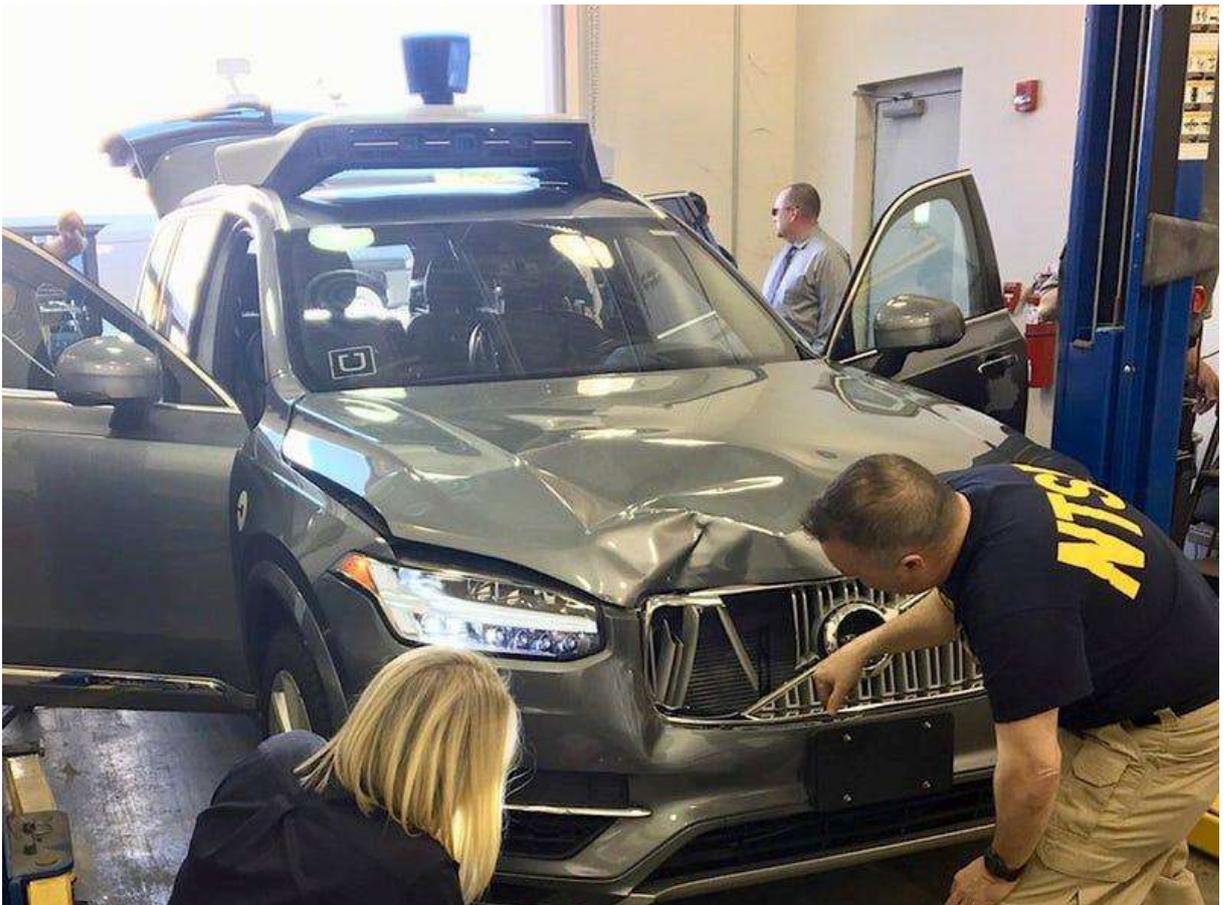
Fonte: World Economic Forum, BCG analysis

Figura 1.1.2 - Porcentagem de pessoas que concordam que carros autônomos não serão seguros.



Fonte: 2020 Global Automotive Consumer Study pág.9

Figura 1.1.3 - Investigadores do Departamento Nacional de segurança de Transporte (NTSB) examinam um Carro Autônomo da Uber que ocasionou um atropelamento fatal em Tempe, Arizona em 20 de março de 2018



Fonte: National Transportation Safety Board/Handout via REUTERS

1.1 MOTIVAÇÃO

O impulso principal deste trabalho é empregar a visão computacional e conceitos de *Deep Learning* na criação de um protótipo de carro autônomo. A cada capítulo vamos explicar o conceito da tecnologia que servirá para o desenvolvimento do protótipo e, sempre que possível, comparar diferentes algoritmos com o objetivo de constatar qual melhor se aplica.

O capítulo 2 apresenta um *overview* do avanço dos carros autônomos na história e seus principais progressos até hoje. No capítulo 3 será explicado o conceito das *smart roads* e suas aplicações. No quarto capítulo será elucidado o conceito de visão computacional e qual sua função no algoritmo. No quinto capítulo o conceito de *Deep Learning* e suas diferentes vertentes de aprendizagem são explanados, além disso mostra como foi feito o setup do algoritmo a fim de torná-lo mais eficiente no projeto.

No capítulo 6 são apresentadas as ferramentas que tornaram possível o desenvolvimento do modelo. Foi reservado um anexo com os detalhes sobre as ferramentas para trazer maiores informações caso o leitor sinta necessidade.

No capítulo 7, com os pilares previamente explicados, será apresentado o modelo, tornando mais fácil a compreensão de todas as nuances na construção do algoritmo e os parâmetros que compõem seu código.

Os resultados obtidos através da experimentação e simulação de cada versão do modelo e seus respectivos aperfeiçoamentos serão apresentados no oitavo capítulo. No capítulo seguinte, será feita a análise e seus impactos no desenvolvimento do modelo. No capítulo 8 será feita a conclusão com base em toda a metodologia apresentada nos capítulos anteriores. No capítulo 9 é apresentado o cenário brasileiro com a mobilidade autônoma e, finalmente, no capítulo 11 podemos discutir futuros trabalhos.

1.2 PROBLEMA A SER RESOLVIDO

A transição do modelo atual, onde a interferência humana é predominante, para o modelo completo de automatização dos carros, no qual não existe nenhuma

interferência humana, será problemática. Já está sendo desenvolvido o modelo de *smart city* e *smart roads* e como funcionariam os carros nesse modelo. Mas e em países onde não há condições de ser aplicado o modelo completo de automação? Como esses carros autônomos se comportariam? O que aconteceria se o carro perdesse a conexão com as *smart roads*? Um dos principais problemas seria a segurança desses carros caso estivessem em situações descritas acima.

1.3 OBJETIVOS

O objetivo principal deste trabalho consiste em desenvolver um modelo de inteligência artificial usando sensores de visão computacional que possam mantê-lo em curso em um determinado trajeto. Para isso, será usado um simulador, onde todas as condições de terreno, ambientação do trajeto, climatização e condições do veículo serão as mais próximas possíveis da realidade. Será construído um modelo de IA para analisar as informações alimentadas pelo sensor da câmera que, por sua vez, irá guiar o veículo respeitando a trajetória e placas de sinalização.

2 A HISTÓRIA DOS CARROS AUTÔNOMOS

No começo do século 20, a humanidade estava espantada com as feiras tecnológicas, muito comuns naquele tempo. Nessas feiras eram apresentadas tecnologias, invenções e ideias que estavam sendo estudadas e poderiam de alguma forma chegar algum dia para o uso diário. O grande atrativo dessas feiras era a visão do futuro que apresentava rodovias inteligentes que guiavam carros “autônomos” que não precisavam que o usuário estivesse no comando do veículo durante o percurso. As multidões que visitavam as feiras se perguntavam se algum dia isso seria possível.

Várias grandes empresas da linha de automobilismo investiram esforços e muitos recursos para poder imaginar uma forma de tentar se aproximar dessa visão utópica do futuro. Muito progresso foi alcançado a partir de pequenos avanços ao longo da história. Podemos mencionar alguns marcos históricos do desenvolvimento do carro autônomo como sendo o de Francis P Houdina, engenheiro elétrico do exército americano e fundador da empresa *Houdina Radio Control* e que teve a primeira demonstração de um carro controlado por rádio controle ("Phantom Auto" to Be Operated Here, 1932).

Outro grande expoente no desenvolvimento de carros autônomos foi o engenheiro Ernst Dieter Dickmanns que liderou o seu time na Universidade *Bundeswehr* e teve muito progresso, com destaque principalmente no campo da visão adaptada em carros. Ele conseguiu criar sensores e usar em Mercedes-Benz fazendo que pudesse ter direção, aceleração e freios controlados por computadores (DELCKER, 2018).

A seguir vamos entender como houve esse desenvolvimento desde o começo até os dias atuais e como podemos imaginar o futuro da mobilidade urbana.

2.1 DESENVOLVIMENTO TEÓRICO

Na feira de Nova York de 1939, a *Futurama*, foi apresentado seu pavilhão *Highways and Horizons*, (Fig.2.1.1 e Fig. 2.1.2) no qual a General Motors mostrava como seria o mundo 20 anos à frente, nesse pavilhão foi apresentado o conceito de estradas inteligentes e superentradas, desenvolvido pelo engenheiro Bel Geddes (KRÖGER, 2016).

Podemos pensar que esse foi o primeiro grande boom no interesse em carros autônomos com grande repercussão mundial. Alguns anos antes, mais precisamente por volta de 1925, foi documentado nas ruas de Nova York, um carro controlado por rádio controle. O inventor Francis P Houdina adaptou um 1926 *chandler* (Fig. 2.1.3) com uma antena de rádio, e o controlou por meio de um carro próximo com uma outra antena transmissora (LUTIN e KORNHAUSER, 2013).

Na década de 50 foi apresentado pela RCA *Labs* um modelo de carro que era guiado por fios inseridos no chão. Essa ideia foi aprimorada pelo engenheiro de trânsito Leland M. Hancock, que resolveu aplicar o conceito apresentado em ruas verdadeiras.

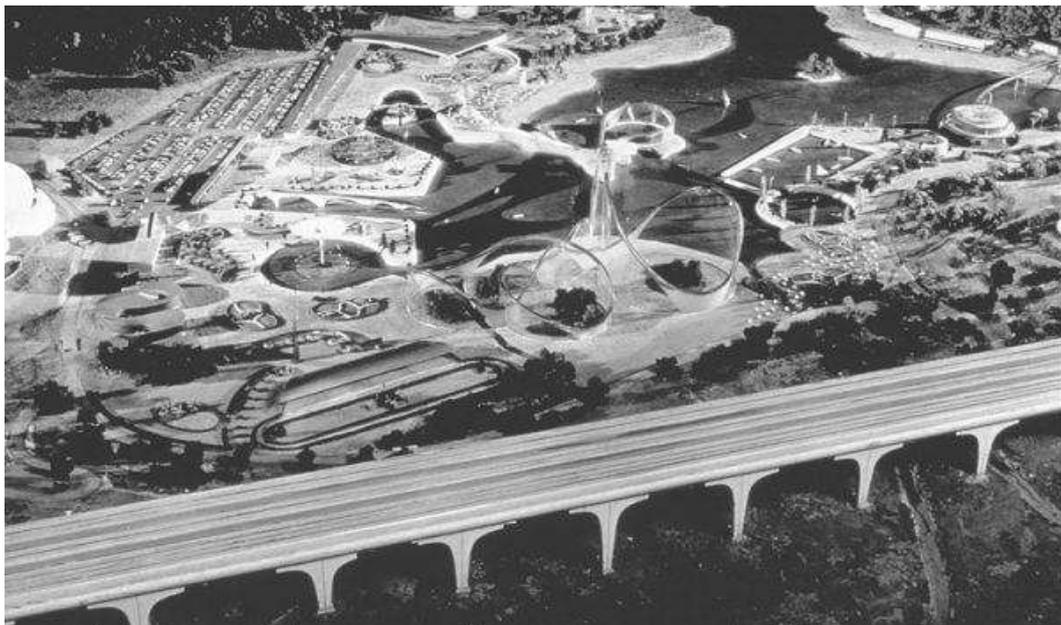
Em 1957, o Estado de Nebraska juntamente com a RCA *Labs* apresenta um trecho da rodovia US 77 e a interestadual 2. Nesse trecho foram implantados sensores nas rodovias que enviavam sinais elétricos para os carros dizendo a posição e a velocidade de objetos metálicos que estavam transitando na rodovia.

Figura 2.1.1 - Pavilhão da General Motors na Feira Mundial 1939 em Nova York



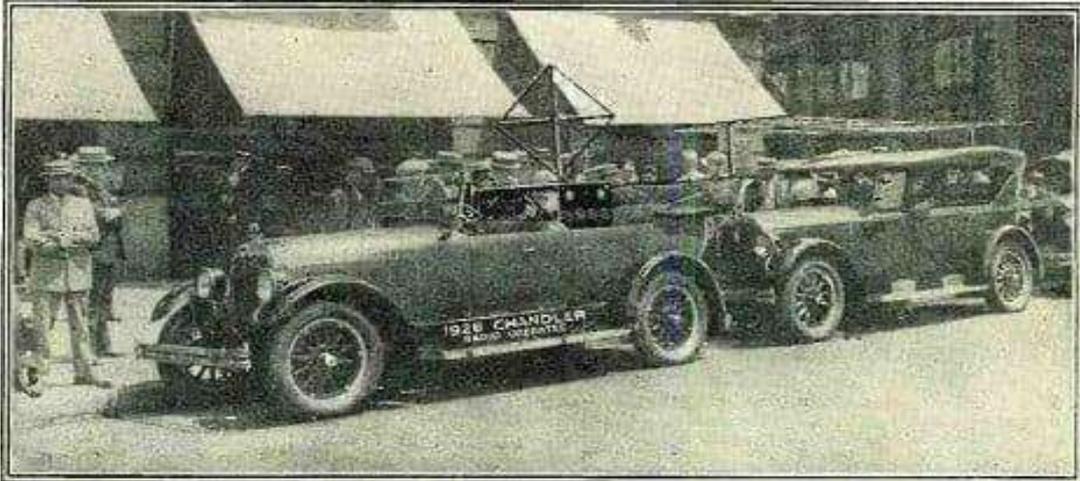
Fonte: Acervo Estadão,2019

Figura 2.1.2 - Pavilhão da Highways and Horizons



Fonte: Acervo Estadão,2019

Figura 2.1.3 - Um 1926 Chandler controlado por rádio controle em 1925 em NY



Fonte: *Discover Magazine*, 2017

Nas décadas seguintes, todos os avanços que houve nos carros, foram essencialmente sensores que ajudavam o usuário na condução. Essa realidade foi mudada quando o exército americano, com intuito em avanços bélicos, promoveu uma competição exclusiva de carros autônomos. A DARPA (*Defense Advanced Research Projects Agency*).

A agência de pesquisa e desenvolvimento militar dos estados foi pressionada, logo após o atentado de 11 de setembro, das torres gêmeas, a reduzirem o uso de soldados americanos em campos de batalhas. Para isso, foi criada uma competição entre as maiores universidades americanas, na qual carros completamente autônomos tinham que percorrer percursos das mais variadas situações e chegar a pontos determinados. O *Grand Challenge* (Fig.2.1.4), tinha como prêmio US\$ 1 milhão para o carro que percorresse 241 quilômetros no deserto de Mojave, localizado entre a Califórnia e Nevada.

Devido a problemas técnicos, nesse primeiro desafio em 2004, não houve vencedor. O carro da Universidade Carnegie Mellon foi o melhor colocado ao completar 5% do percurso total da prova. A DARPA dobrou o prêmio para US \$2 milhões, assim, 5 carros de 195 participantes conseguiram concluir o percurso inteiro. O carro vencedor foi da universidade de Stanford, os alunos usaram um carro da Volkswagen adaptado em um robô, usando os mesmos recursos do ano anterior. Mas mudaram de abordagem para resolver o problema, assim, houve a evolução do

software em relação ao ano anterior. O carro conseguiu completar o percurso de 212 quilômetros em 6 horas e 53 minutos (AMATO, 2018).

Figura 2.1.4 - O Carro da Universidade Stanford concluindo o percurso no DARPA 2005



Fonte: DARPA ,2009

A US National Highway Traffic Safety Administration (NHTSA) define atualmente carros autônomos em 6 categorias. Como é mostrado na figura a seguir:

Figura 2.1.5 - Classificação dos carros autônomos



Fonte: NHTSA ,2019

No nível zero encontram-se basicamente os carros que não têm nenhuma assistência na direção. No nível 1, os carros já apresentam de uma forma básica alguma assistência, como por exemplo, controle de velocidade, mas que no final, é o motorista que pode usar ou não a assistência.

No nível 2, poderíamos dizer que nesse nível, existe uma automação parcial, onde o carro consegue manobrar e frear sem ajuda do motorista, entretanto, ainda é preciso que o motorista esteja presente para assumir o controle no percurso. Podemos dizer que estamos nesse nível de automação em nível comercial.

No nível 3, muitas montadoras já têm protótipos com esse nível de complexidade, onde o carro consegue estar em modo autônomo boa parte do percurso, mas o motorista tem que estar presente e atento ao percurso para que, caso haja necessidade, assuma o controle. No nível 4, o carro consegue ser autônomo em quase todas as condições, deixando ao motorista a possibilidade de não precisar assumir o controle.

No nível 5, o carro é capaz de dirigir em todas as condições possíveis sem a necessidade de ajuda humana (ANDERSON, KALRA, *et al.*, 2016).

Muitos cientistas estimam que para conseguirmos chegar ao nível 5 de automatização dos carros, teremos que adaptar estradas e cidades. O conceito de *smart roads* (PINDARWATI, 2016) (SMART TRANSPORTATION ALLIANCE, 2015) é muito conhecido desde o começo dos estudos de carros autônomos. Muitos especialistas afirmam que o conjunto de carros autônomos, *smart roads* e *smart cities* é a ponte para um futuro de grande transformação em nossa sociedade. Podemos esperar transformações no campo da logística de produtos e de mobilidade urbana, serviços serão descontinuados por motivos de obsolescência de novas tecnologias como por exemplo seguros contra acidentes de carro (CAPGEMINI RESEARCH INSTITUTE, 2018).

Mas como podemos esperar a transição dessas mudanças? Uma das tecnologias que estão dando grande vazão para esse futuro é a Internet das Coisas. Com aparelhos cada vez mais conectados e produzindo dados em tempo real, não é de espantar o alto nível de precisão que carros e devices irão trocar entre si. A 4ª Revolução Industrial está baseada principalmente em análise de dados (BLOOM, TAN, *et al.*, 2017) ou os dados que esses mesmos aparelhos coletam e transmitem

para a internet. Todas essas informações ajudaram carros autônomos a tomarem decisões mais precisas e em um menor tempo (BLOOM, TAN, *et al.*, 2017).

Existe um longo caminho pela frente, teremos que discutir e entender melhor sobre permissões e legislações sobre carros autônomos. Mas todos podemos concordar de antemão quanto à segurança que haverá quando tivermos feito a transição por completo (GOUVÊA, 2016).

3 SMART ROADS

Um dos grandes problemas da economia se encontra na logística e nos meios de transporte. Existe uma pressão muito grande para que as rodovias se tornem cada vez mais seguras e eficientes, e para que isso aconteça, uma infraestrutura mais inteligente é necessária. As cidades inteligentes ou *Smart Cities* estão sendo desenvolvidas e construídas cada vez mais usando tecnologias de IoT (*Internet of Things* - Internet das Coisas) com isso é possível coletar e analisar uma quantidade maior de dados, melhorar o gerenciamento diário do tráfego das rodovias, conseguir se adaptar e se planejar para futuras mudanças nas necessidades logísticas.

Com sensores IoT, câmeras com alta definição, radares e transmissões 5G, os dados podem ser analisados quase em tempo real e, assim, melhorar o tráfego em rodovias inteligentes. Os dados podem ser armazenados em cloud para análises mais profundas e planejamentos futuros, sendo úteis para dados críticos como emissões de CO₂ ou melhorias nas condições de tráfego (TOH, SANGUESA, *et al.*, 2020).

Com avanços computacionais, abre-se uma gama enorme de possibilidades para rodovias inteligentes e mais conectadas. Proporciona latência baixa para análise de dados e uso de Inteligência Artificial na infraestrutura das rodovias inteligentes. Um bom exemplo seria o de semáforos adaptativos que podem ajustar automaticamente seu tempo baseado nos dados coletados pelo sistema de SR, melhorando assim o tráfego ou enviando dados das condições de cada trecho da rodovia diretamente para os SDC.

As principais chaves tecnológicas que estão por trás do avanço das smart roads são: conectividade, computação em cloud, análise de dados, sensores, *internet of things* e inteligência artificial. As rodovias inteligentes têm uma variedade muito grande de aplicações usando as tecnologias citadas anteriormente.

Transporte é a base principal da economia global. A logística de bens de consumo e de seres humanos ajudou no desenvolvimento de economias fortemente conectadas globalmente e o aparecimento de novas cidades. Com o avanço da sociedade, houve uma explosão no aumento de meios de transporte, provocando, por sua vez, aumento nos problemas de transporte, podemos citar algum deles como congestionamento nas rodovias, acidentes, poluição, aumento do custo de combustível, escassez de combustível e aumento do custo de apólice de seguros. Apesar de ser um problema de engenharia civil e engenharia de transportes, apenas com o conceito de *smart roads* será possível diminuir ou até mesmo resolver definitivamente muitos desses problemas.

No futuro previsto por vários cientistas, podemos destacar que a super velocidade da transmissão de informação é a base para o conceito de rodovias inteligentes. As rodovias não são mais vistas com uma visão tradicional de uma entidade física ou apenas um chão sólido, mas estão sendo transformadas com características (Tab. 3.1.1) que não seria possível no passado como transmissão de dados em tempo real de cada trajetória. Mas muito ainda está por ser pensado e desenvolvido, mas os primeiros passos já foram dados, existem vários grupos, consórcios como por exemplo: e ISO TC 204, IEEE 802.11WAVE para a padronização da tecnologia que será adotada globalmente.

Figura 2.1.1 - *Smart Roads* pode se comunicar entre as rodovias e usuários igual a um objeto conectado na rede da internet com atualizações de informações



Fonte: <https://www.linkedin.com/pulse/what-smart-road-luigi-carrarini/>

3.1 RODOVIAS QUE PRODUZEM ENERGIA

O objetivo das rodovias sempre foi levar de um ponto a outro ponto. Contudo, no modelo das rodovias inteligentes, elas farão muito mais do que ser um meio pavimentado para veículos. Uma das ideias que surgiu foi a coleta e distribuição de energia pelas rodovias (Clean Energy Production through Roadway Surface Power Harvesting Systems for Increased Safety in the Transportation Sector, 2013).

Atualmente existem várias formas de se conseguir extrair energia através das rodovias. Uma forma de captação seria o uso do sol como fonte primária, essas rodovias são chamadas de *Solar Roads* (Tab.3.1.3). Outros modelos usam trepidações mecânicas ocasionadas pelos veículos produzindo energia elétrica. A energia capturada pode ser usada para alimentar a rodovia, como por exemplo: iluminação rodoviária, sinalização e semáforos ou pode retornar para a rede elétrica.

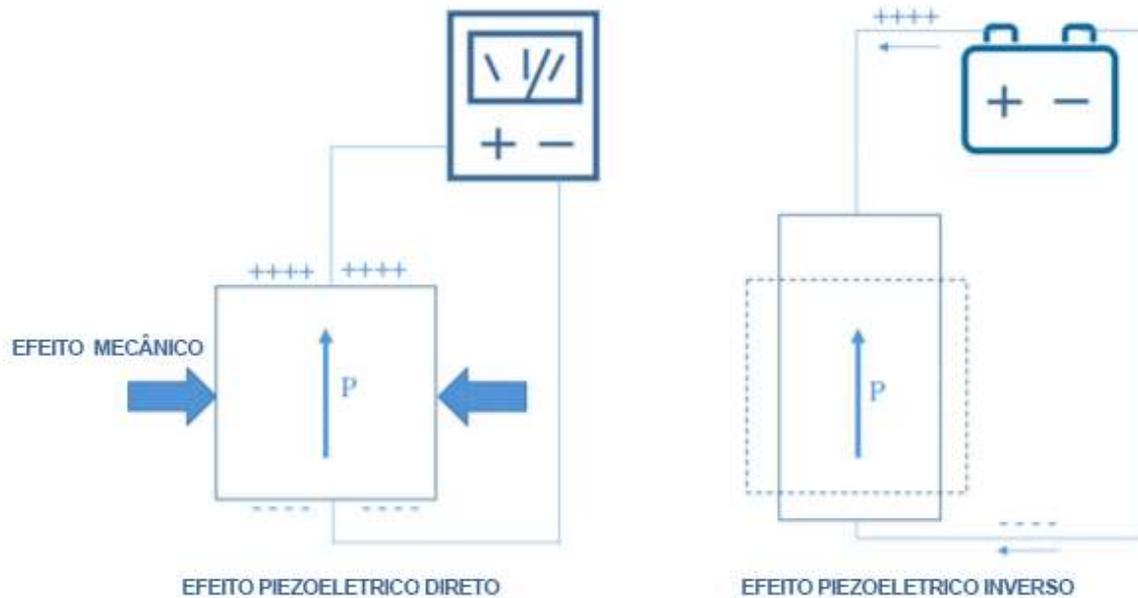
Rodovias piezoelétricas (HILL, AGARWA e TONG, 2014) usam aparelhos piezoelétricos para gerar energia elétrica. Os cristais piezoelétricos são colocados abaixo da superfície do asfalto e esses cristais sofrem uma deformação mínima quando veículos passam sobre eles. Essa deformação produz uma corrente elétrica devido ao efeito piezoelétrico do cristal (Fig.3.1.1) (Tab.3.1.2).

Tabela 3.1.1 - Resumo dos 10 avanços mais significativos nas rodovias inteligentes

número	avanços	setor
1	Rodovias que produzem e estocam energia	Energia
2	Rodovias que cantam: <i>musical road</i>	Segurança
3	Rodovias que pesam veículos	Logística
4	Rodovias que carregam: <i>electrified roads</i>	Energia
5	Rodovias com sinais de tráfego sem fio	Segurança
6	Rodovias que detectam violações de trânsito	Leis de trânsito
7	Rodovias que usam 'comunicação' (V2X)	Segurança, logística e leis de trânsito
8	Rodovias com cruzamentos inteligentes	Segurança
9	Rodovias com resgate de emergência rápido	Segurança
10	Rodovias com semáforos inteligentes	Serviços urbanos

Fonte: *Advances in smart roads for future smart cities*

Figura 3.1.1 - Característica do funcionamento do aparelho piezoelétrico



Fonte: *Advances in smart roads for smart cities*

Tabela 3.1.2 - Desenvolvimento e aplicação de rodovias piezoelétricas no mundo

País	Rodovias piezoelétricas
Estados Unidos	Investido US\$2.3 milhões em projetos relacionados em rodovias piezoelétricas; A Comissão de Energia da Califórnia e do Oregon são as que estão liderando a iniciativa. Em Washington, DC, 240 metros quadrados foram implantados pavimentos energizados pelo andar humano.
Inglaterra	O programa de pesquisa da Universidade de Lancaster está gerando energia pelo tráfego na rodovia, com um objetivo de geração 1-2 MW/km no volume do tráfego de 3000 carros/h. <i>Bird's street</i> , Londres central, rodovias piezoelétricas instaladas alimentam as luzes na rodovia.
Japão	Companhia Ferroviária do Leste do Japão – 2008-2009 lançamento na estação Marunouchi na saída norte em Tóquio. Produz eletricidade alcançando 10K Watt-segundo por dia. A eletricidade gerada é usada para alimentar todos os dispositivos eletrônicos da estação.
Itália	Rodovia Piezoelétrica de Veneza até Trieste.

Fonte: *Advances in smart roads for smart cities*

Tabela 3.1.3 -Desenvolvimento de rodovias solares na China, França e Estados Unidos

País	Rodovias Solares
China	Província de Jinan, Shandong – 2km de rodovias com módulos solares
França	Rodovia solar de Wattay alimenta praças de pedágios
Estados Unidos	Solar Roadways é uma startup americana que trabalha com a geração de eletricidade de rodovias e substituindo concreto e asfalto das rodovias por painéis solares (MOUYAL, 2018). Cada quadrado é um painel solar de 44W protegido com vidro temperado, capaz de aguentar o peso de um caminhão. As luzes de LED servem como marcação nas linhas. Os quadrados ou painéis são modulares e podem ser facilmente substituídos durante a manutenção.

Fonte: Mouyal, 2018

3.2 RODOVIAS QUE PRODUZEM MÚSICAS

As rodovias musicais (Fig.3.2.1) produzem música e sons quando os veículos passam sobre elas. Esse modelo é muito usado em países como Estados Unidos, Japão, Dinamarca, Holanda e Coreia do Sul. O Instituto Industrial de Pesquisa de Hokkaido desenvolveu superfícies que produzem músicas nas rodovias, onde os carros são os instrumentos que fazem tais músicas. O conceito por trás das rodovias musicais é o de ranhuras colocadas em intervalos de espaço entre elas nas rodovias para produzirem sons musicais. Dependendo do espaçamento entre elas e o tamanho das ranhuras, os carros conseguem variar a tonalidade do som ao passar por elas.

O propósito principal das rodovias musicais é a segurança e não somente o entretenimento. Esse artifício ajuda motoristas a se manter em curso nas rodovias, além de prever acidentes como excesso de velocidade e fadiga no trajeto. Um problema desse modelo é o ruído. Residentes que moram perto dessas rodovias, podem ser constantemente acordados pelas músicas da rodovia. Para evitar essa inconveniência, esse modelo de rodovia é usado longe de residentes, são comumente usados em trajetos nos quais o motorista deve ser lembrado de controlar sua velocidade (Tab.3.2.1).

Figura 3.2.1 - Rodovia Musical de Lancaster na Califórnia

Fonte: *California.org*

Tabela 3.2.1 - Implantação do uso das rodovias músicas ao redor do mundo

País	Principais rodovias musicais
Japão	O Japão tem mais de 30 rodovias musicais, localizadas em Hokkaido, Hiroshima, Shizuoka, Oita, Gunman, etc.
Coréia do Sul	' <i>Mary had a little lamb</i> '—para avisar motoristas nas rodovias coreanas.
Taiwan	Localizado em <i>King Men 's 'Dinglin Road</i> ', carros com velocidade até 50 kph produzem a música ' <i>The Olive Tree</i> '. Tiras de fita de anti-derrapagem são usadas ao invés de cortes permanentes no asfalto.
Dinamarca	Em 1995, artistas dinamarqueses inventaram o ' <i>Asphaltophone</i> ', marcas salientes no asfalto que produzem músicas. O propósito é manter os motoristas acordados e não ultrapassar o limite de velocidade.
Holanda	Em Leeuwarden, a rodovia vai tocar o hino regional de <i>Friesland</i> com o motorista passando com velocidade até 40 mph.
Estados Unidos	New Mexico - Na rota histórica 66 entre Albuquerque e Tijeras, a música ' <i>America the Beautiful</i> ' pode ser ouvida por motoristas que passam com velocidade até 45 mph. Lancaster, CA - Um trecho de ' <i>William Tell Overture</i> ' toca para motoristas que estejam até 55 mph.

Fonte: *Chai K. Toh, 2020*

3.3 RODOVIAS QUE CARREGAM AUTOMATICAMENTE VEÍCULOS

Uma das grandes preocupações é a forma como vamos lidar com a poluição desenfreada em que vivemos. A grande aposta para resolver esse problema seria carros elétricos, o único problema que apresenta é a autonomia das baterias. É muito difícil achar estações de carregamento quando a bateria está com nível baixo de energia, especialmente em regiões rurais ou afastadas de áreas urbanas. Na Coreia do Sul, ônibus que são usados em campos universitários são carregados automaticamente e continuamente, fazendo com que não necessite de uma bateria tão grande e diminuindo o peso total do veículo (HOSTER, 2017).

Muitas cidades estão sugerindo o uso de uma linha específica da rodovia para o carregamento de carros autônomos. Caminhões que fazem a logística de produtos em longas distâncias, precisam constantemente ser recarregados para manter o horário esperado. Na Suécia, aproximadamente 1,5km perto de Stockholm foi transformado em rodovias elétricas. Esse trecho carrega carros e caminhões que transitam nela. Isso faz parte de um projeto nacional chamado '*eRoadArlanda*' (Fig.3.3.1) onde o objetivo principal é promover carregamento pelas rodovias enquanto estão em movimento, ao invés de criarem estações de carregamento ao lado da rodovia a cada intervalo de distância (World's first electrified road for charging vehicles opens in Sweden, 2018).

Na Grã-Bretanha, o governo está muito empenhado na tecnologia de rodovias e carregamento de veículos elétricos (Fig.3.3.2). Uma das tecnologias que estão sendo desenvolvidas é o carregamento por indução magnética. A ideia principal é instalar cabos subterrâneos nas rodovias que irão gerar campos eletromagnéticos potentes o suficiente para acionar o receptor nos veículos elétricos e, assim, acionar o carregamento (KOTTASOVA e PETROFF, 2015). Essa tecnologia usa o princípio da lei eletromagnética de Faraday.

Figura 3.3.1 - Instalação de linhas de carregamento para veículos elétricos do projeto *eRoadArlanda* na Suécia. O governo tem um planejamento de expansão nacional do projeto



Fonte: *The Guardian*, 2021

Figura 3.3.2 - Rodovias Elétricas que carregam veículos elétricos enquanto estão na linha verde de carregamento, sem precisar parar o percurso



Fonte: *Money.com*, 2021

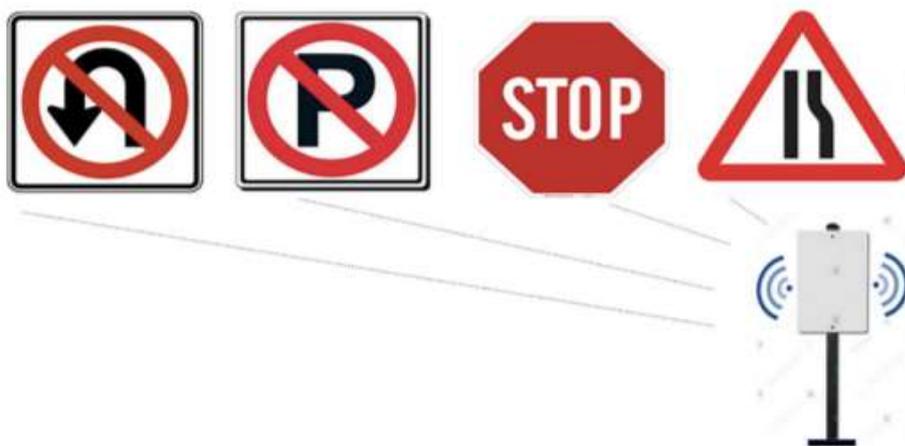
3.4 RODOVIAS COM SINAIS DE TRÂNSITO SEM FIO INTELIGENTE

O uso de sinais de trânsito foi um grande avanço para aumentar a segurança no trânsito. Existem mais de 60 sinais na legislação americana e mais de 170 na legislação inglesa. Isso dificulta muito a memorização, especialmente para motoristas idosos e com problemas de memorização.

Como mostra a figura 3.4.1 a seguir, o poste transmite sinais através de comunicação via 5G informando condições sobre determinado trecho da rodovia para o recipiente do carro autônomo que ao receber os dados atualiza-os para seu atual estado no determinado trecho da rodovia (TOH, CANO, *et al.*, 2019). Esse conceito pode ser usado tanto para carros autônomos como para motoristas humanos. A vantagem para motoristas humanos é a não necessidade do motorista ter que buscar ao longo da rodovia sinais que possam lhe informar, diminuindo riscos de acidentes por desatenção.

As vantagens do uso dessa tecnologia são inúmeras, podemos citar algumas mais significativas como por exemplo: não há necessidade do sinal ser visível ao olho humano, desobriga motoristas da necessidade de buscar por sinais ao longo da rodovia, o motorista não precisa lembrar de todos os sinais na legislação vigente, não é afetado por visibilidade climática, os sinais são programados e, por isso, podem ser facilmente alterados e atualizados, e o mais importante, não há necessidade de reconhecimento visual através de câmeras pelos carros autônomos.

Figura 3.4.1 - Conceito do Poste programável de sinais de trânsito

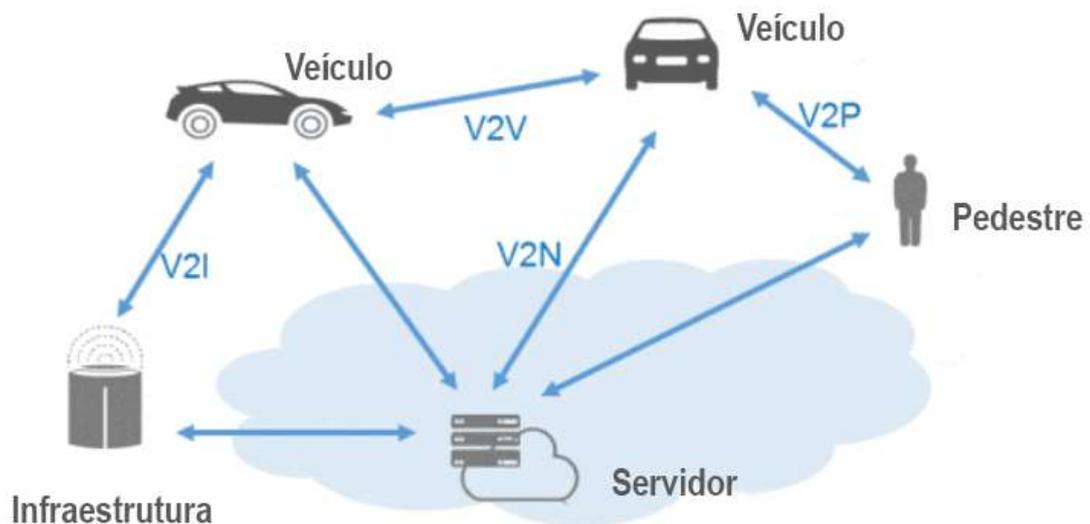


Fonte: Chai K. Toh, 2021

3.5 RODOVIAS COM V2X E VANETS

Com o avanço do protocolo de transmissão e o desenvolvimento da internet das coisas, foi possível aplicar para a parte de mobilidade. Um dos pilares do futuro da mobilidade está na parte em como os carros autônomos vão se comunicar entre si, trocando dados em tempo real. Essa forma de comunicação entre carros autônomos tem o nome de Veículo-Para-Veículo ou V2V, como é comumente conhecido (Fig.3.5.1). Outro tipo de comunicação seria do veículo para a rodovia, essa forma é chamada de V2I e a comunicação entre o veículo e algum outro objeto é chamado de V2X. Esse tipo de comunicação é muito importante para manter a segurança do veículo alertando sobre sua posição, velocidade e perigos em potencial ao seu redor. Isso é possível graças às informações que o SDC envia e recebe com a estrutura da rodovia onde circulam, veículos que estão ao seu redor e objetos, garantido uma melhor árvore para tomada de decisão (MINGHUANG, SHUCHIANG, *et al.*, 2017) (Tab.3.5.1).

Figura 3.5.1 - Tipos de comunicação dentro da arquitetura V2X



Fonte: Autor

Tabela 3.5.1 - Diferentes tipos de comunicação veicular e suas características

Tipo	Características e Aplicações
V2I	Exemplos: 802.11n, LTE Regularização do fluxo de trânsito Cobrança de pedágio
V2V	ITS 5.9 GHz spectrum IEEE 802.11p Comunicação de alerta e mensagens curtas Propagação de aviso de perigo Evitar colisões
V2N	Exemplo: celular LTE 4G / 5G Veículo-para-network; requisição para assistência ao acesso a banco de dados na cloud e extrair informações pertinentes.
V2X	Combinações de V2V, V2I, V2E (ambiente), V2P (pedestre), V2C (cloud), V2H (Home) Indica veículo para qualquer coisa (X) Não precisa de rede celular para ser usada.
Celular – V2X	3GPP lançou 14 C-V2X padrão, baseado em LTE Completamente compatível com tecnologias 5G Posicionamento preciso, disponibilidade em condições de tráfego para dirigir ou direção completamente autônoma. Suporte para alta densidade de comboio veicular. Suporte para evitar colisão. Disponibiliza alta banda de comunicação de dados.

Fonte: Chai K. Toh, 2021

3.6 RODOVIAS COM INTERSECÇÕES INTELIGENTES

As intersecções são uma das grandes causas de acidentes por vários motivos, como por exemplo a obstrução visual de veículos que vêm de várias direções. Sabendo disso, vários centros de pesquisas têm estudado por anos uma forma de atenuar ou quem sabe reduzir a zero o número de acidentes em intersecção. Existe

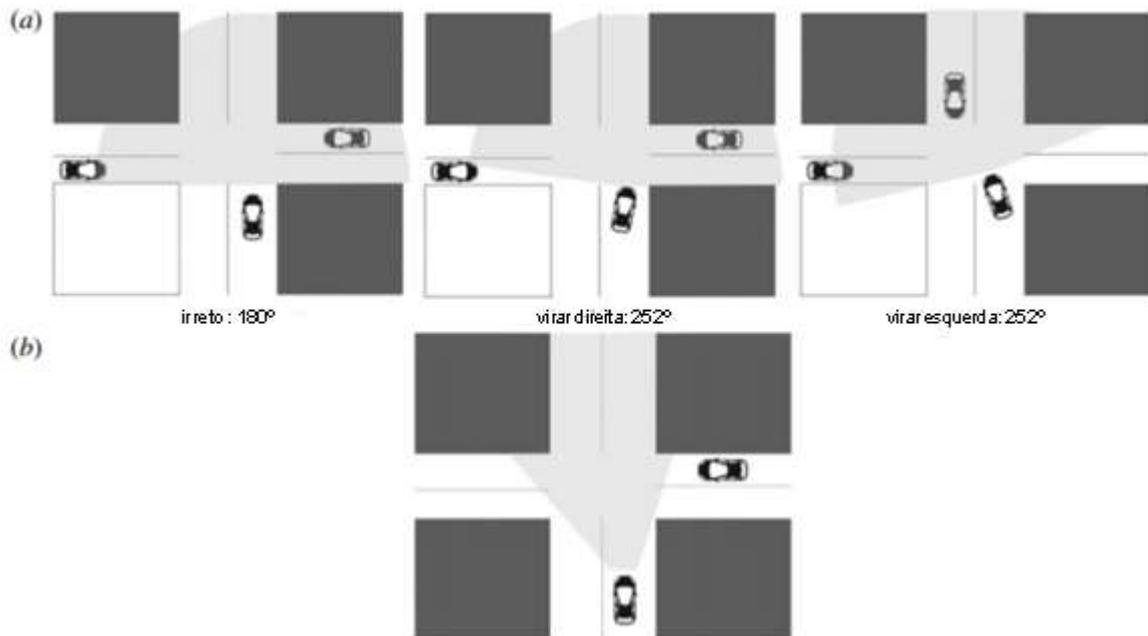
um projeto de cooperação europeu (RÖSSLER, 2007) para desenvolver tecnologias em intersecção. A intersecção inteligente usa todas as informações coletadas em várias tecnologias como sensores em SDC, software de reconhecimento de objetos e comunicação V2I (Veículo para Infraestrutura). Isso resulta em um mapeamento detalhado de acidentes e melhora a redução de acidentes em intersecções. Com intersecções inteligentes é possível obter análise detalhada e compreensível de acidentes, identificar cenários comuns onde acontecem acidentes e os tipos de intersecções que mais causam acidentes dentre países europeus. O sistema inteligente de intersecção divide objetos em três categorias, a saber: veículos estacionados, veículos em movimento e pedestres.

No projeto de segurança das intersecções inteligentes, os VRUITS (ITS o para VULNEAREABLE ROAD USERS), mensagens de aviso são transmitidos para os usuários da rodovia (pedestres, ciclistas e motoristas). No sistema que a empresa Miovision (Miovision TrafficLink hardware, 2021) desenvolveu, a intersecção inteligente consegue entender o tráfego e disparar respostas em tempo real. Algumas dessas respostas podem ser: estender o tempo de luz verde para que ciclistas possam atravessar com segurança; identificar pedestres com dificuldades visuais ou idosos com o intuito de alertar carros autônomos e carros comuns; dar prioridade para carros de emergências como ambulâncias e carros de polícia ao atravessar; identificar e analisar o fluxo de pedestres em tempo real e, dessa maneira, criar tempo de respostas nos intervalos do semáforo a fim de reduzir o tempo das luzes do semáforo e ajustar o fluxo em tempo real.

No projeto proposto pela Honda para intersecções inteligentes, câmeras são instaladas nas esquinas que formam a intersecção como visto nas figuras 3.6.1 e 3.6.2. Um software de processamento de imagem é usado para criar uma imagem 360° da intersecção. Sistemas de inteligência artificial e reconhecimento de objetos são usados para classificar objetos em pedestres, ciclistas, motoqueiros, veículos de emergência e veículos comuns. Essas informações são transmitidas para carros que se aproximam da intersecção auxiliando na tomada de decisões. Com isso, diminui-se o risco de pontos cegos na intersecção.

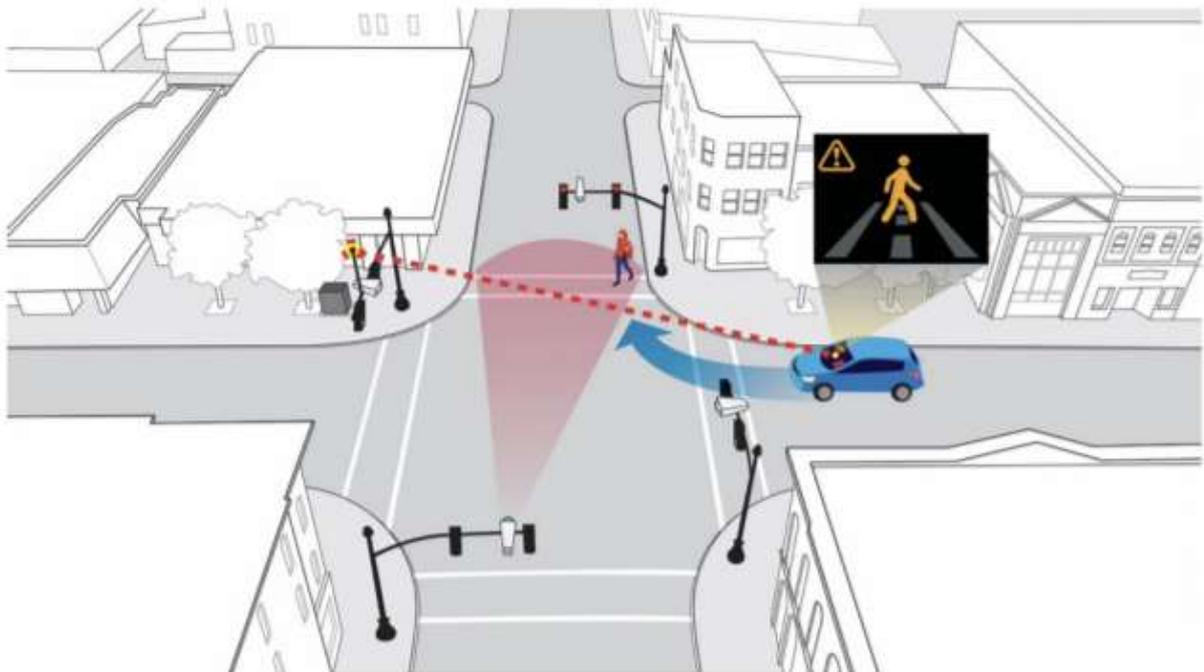
Com o progresso, a tecnologia está se tornando mais acessível, o sistema de intersecção inteligente está incluso nos conceitos das *smart roads* e *smart cities*, as quais são poderosas ferramentas na redução de acidentes, que proporcionam mais segurança aos pedestres.

Figura 3.6.1 - Em (a) os possíveis acidentes que podem ocorrer na intersecção e em (b) o ponto cego do motorista em uma intersecção.



Fonte: Chai K. Toh, 2021

Figura 3.6.2 - Exemplo de uma intersecção inteligente que usa câmeras com reconhecimento de objetos e comunicação V2X para avisar SDC antecipadamente



Fonte: Honda News, 2018

Tabela 3.6.1 - Exemplos de aplicação de intersecções inteligentes ao redor do mundo

País	Comunicações
Estados Unidos	Detroit, USA: Na cidade de Detroit, mais de 40% das intersecções têm sido instaladas com o sistema de Intersecções inteligentes desde junho de 2018. É considerado o primeiro sistema inteligente de intersecção no mundo, com sensores, câmeras conectadas aos sinais de tráfego e monitoramento remoto. O principal objetivo do sistema é aumentar a segurança nas intersecções. O sistema consegue gerar dados de acidentes e a Inteligência artificial consegue prever e evitar futuros acidentes.
Japão	Em 2018, A Honda anunciou o Sistema de intersecção inteligente por comunicação V2X desenvolvido para reduzir acidentes de tráfego nas intersecções. Uma demonstração desse sistema com sucesso foi feita na cidade de Marysville, Ohio. O sistema permite que motoristas consigam ver virtualmente através e ao redor de prédios e muros com software de reconhecimento de objetos e câmeras instaladas em toda a intersecção e com comunicação V2X.

Fonte: Chai K. Toh, 2021

4 VISÃO COMPUTACIONAL

Neste capítulo vamos estudar um pouco sobre visão computacional, um dos pilares centrais desta dissertação. Para entender o que seria necessariamente o termo visão computacional, precisamos primeiro entender como uma imagem pode ser formada e compreendida no âmbito digital.

Quando um ser humano dirige, ele usa sua visão para poder reconhecer a rua, carros, placas, pessoas e objetos em geral ao longo do percurso. Mas os carros autônomos ou SDCs (*Self Drive Car*) precisam usar câmeras que possam representar como a visão humana obtém as informações necessárias para o trajeto.

Segundo a definição dos pesquisadores Dana Ballard e Chris Brown em seu mais famoso livro sobre o assunto, a definição de visão computacional seria: “Ciência

que estuda e desenvolve tecnologias que permitem que máquinas enxerguem e extraíam características de imagens (BALLARD e BROWN, 1982).

A visão pode ser dividida em várias etapas. A aquisição da imagem, o processamento da imagem, a extração de características e a detecção de padrões.

Toda imagem, é vista como uma matriz onde cada elemento da matriz, seria o pixel e ele é representado com sua variação numérica de cinza. Veja o exemplo de uma imagem convertida para padrões matriciais:

Figura 3.6.1 - Conversão da imagem digital

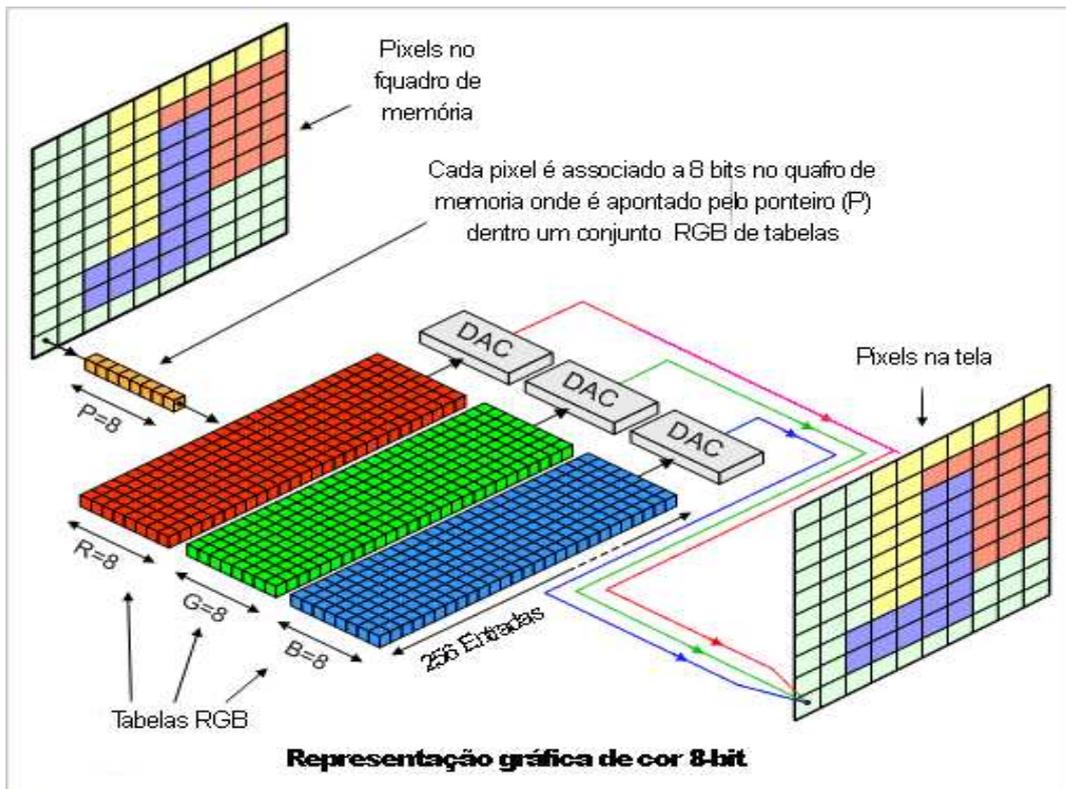


Fonte: Analytics Vidhya 2019

Como podemos verificar, qualquer imagem pode ser vista como uma matriz onde existem variações de preto e branco. A cor preta é representada como 0 e a cor branca como 255.

O mesmo princípio é estendido para imagens coloridas. Para isso é usado o padrão de cores RGB (*RED, BLUE, GREEN*). Neste caso, cada elemento da matriz é formado por uma tríade de valores, que representa os valores de *Red, Blue* e *Green*. Podemos verificar como seria a decomposição de uma imagem colorida em uma matriz (Fig.3.6.2).

Figura 3.6.2 - Decomposição de imagem colorida em matrix RGB

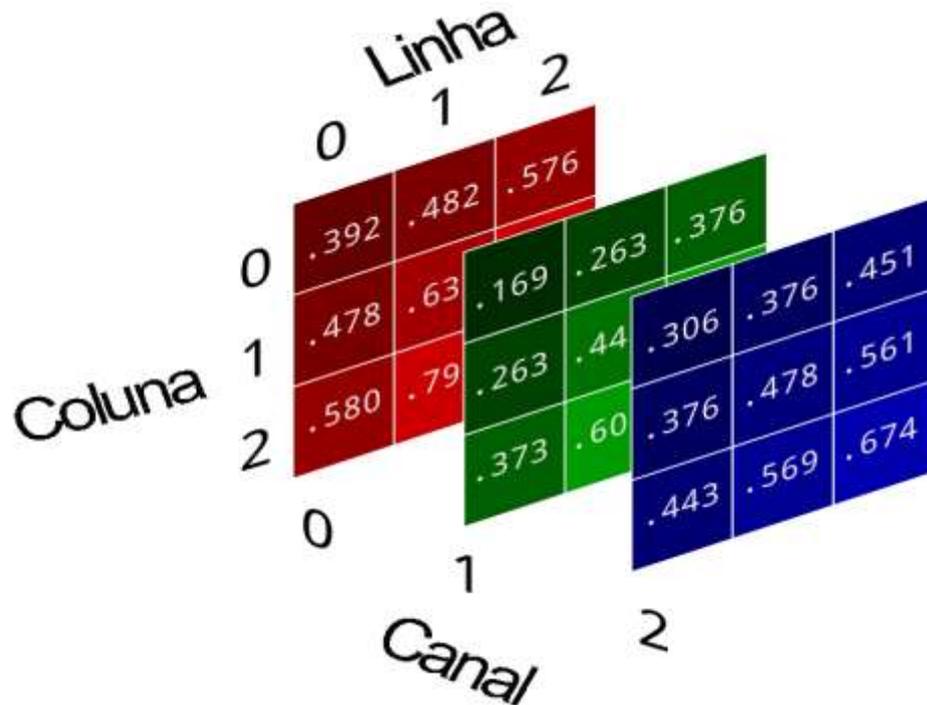


Fonte: Clive Maxfield 2006

O menor elemento é representado como sendo pixel, e é composto por três valores (Fig. 3.6.3). Esses valores representam todas as variações ou combinações do vermelho, azul e verde.

Para a facilitação de análise de imagem, como a extração de características, é aconselhável que possamos converter a imagem colorida para tons de cinza a fim de facilitar o processamento da imagem. A seguir podemos ver como uma imagem colorida é decomposta em uma matriz com a representação do RGB do pixel.

Figura 3.6.3 - Representação matricial de imagem colorida



Fonte: [brohrer.github.io](https://github.com/brohrer)

4.1 FRAMEWORKS: OPENCV

O OpenCv (Open Source Computer Vision) é uma biblioteca *open source* criada pela Intel, muito utilizada para tratamento de imagens e vídeos. Nela existe uma coleção de funções, aproximadamente 2500 algoritmos pré-configurados e otimizados, que podem ser facilmente acessados e contribuem para a manipulação de imagens.

Essa biblioteca foi criada com diferentes propósitos. Entre eles estão: monitoramento de movimentos de objetos, extração 3D de modelos de objetos; extração de pontos 3D de câmeras estéreo; junção de múltiplas imagens para composição e reconstrução de uma cena; busca de imagens em base de dados de imagens; remoção de olhos vermelhos de imagens e outros tipos de remoção por filtros; acompanhamento do movimento dos olhos; reconhecimento de cenários;

reconhecimento de pontos de referência em realidade aumentada, dentre várias outras aplicações.

Por ser uma poderosa biblioteca, é usada tanto em âmbito comercial como no campo de pesquisa em grandes universidades. A seguir vamos fazer um apanhado geral dos principais conceitos dentro da visão computacional e como é o acesso à biblioteca do OpenCV.

A função `cvtColor` (Fig.4.1.1 e Fig.4.1.2) converte a cor de uma imagem para outra seguindo a sintaxe `cvtColor(src, code[, dst[, dstCn]])`. Os parâmetros do argumento dessa função são: `src` é a imagem fonte que vai ser mudada, `code` é modo de conversão, `dst` é a imagem de saída com o mesmo tamanho da imagem de entrada, sendo este último um parâmetro opcional. O `dstCn` refere-se ao número de canais que a imagem de destino vai ter, se o número for zero significa que o número de canais da imagem de saída será igual ao número de canais da imagem de entrada. Esse é o método de retorno de uma imagem. Para entender um pouco como é feita essa conversão:

Equação 4.1.1 - Operação matricial de conversão de cores

RGB[A] to Gray: $Y \leftarrow 0.299 * R + 0.587 * G + 0.114 * B$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

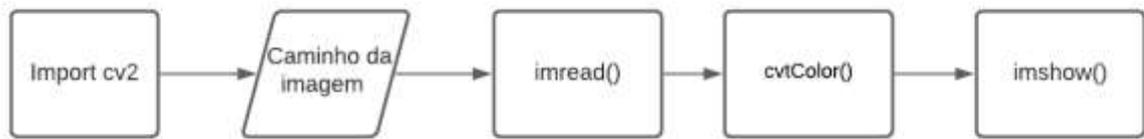
Fonte: Autor

Figura 4.1.1 - Imagem original sem tratamento do método `cvtColor()`



Fonte: Autor

Figura 4.1.2 - Diagrama da função cvtColor()



Fonte: Autor

Figura 4.1.3 - Resultado da imagem após o uso do método cvtColor()



Fonte: Autor

A imagem pode ter ruído, como qualquer sinal. Para amenizar esse ruído, podemos usar um filtro pré-configurado na biblioteca do OpenCV chamado de Filtro Gaussiano ou `gaussianblur()` (Fig. 4.1.5).

A sintaxe da função pode ser vista a seguir `dst = cv2.GaussianBlur(src, ksize, sigmaX[, dst[, sigmaY[, borderType=BORDER_DEFAULT]]])`.

`Dst` é a imagem de saída, `src` é a imagem de entrada ou imagem original, `ksize` indica o tamanho do kernel gaussiano. Os valores são computados através de valores de `sigma`, `sigmaX` é o valor da desviação do kernel padrão ao longo do eixo X, `sigmaY` é o valor da desviação do kernel padrão ao longo do eixo Y e, se o valor de `sigmaY` for zero, então o `sigmaX` é usado para `sigmaY`. O último parâmetro é o `borderType` que especifica os limites da borda da imagem, a seguir estão os possíveis valores:

- cv.BORDER_CONSTANT
- cv.BORDER_REPLICATE
- cv.BORDER_REFLECT
- cv.BORDER_WRAP
- cv.BORDER_REFLECT_101
- cv.BORDER_TRANSPARENT
- cv.BORDER_REFLECT101
- cv.BORDER_DEFAULT
- cv.BORDER_ISOLATED.

Figura 4.1.4 - Diagrama da função de Blur Gaussiano do OpenCV



Fonte: Autor

Figura 4.1.5 - Imagem original à esquerda e a imagem à direita com a suavização do filtro Gaussiano



Fonte: *OpenCV.org, 2021*

A binarização da imagem aborda o uso da função *thresholding* baseado na intensidade de cada pixel. Normalmente a binarização da imagem é realizada após a conversão de escalas coloridas para escalas de cinza na imagem.

Resumidamente, se a intensidade da imagem original for maior que o *threshold* então o pixel correspondente na imagem de saída é marcado como branco. Se a intensidade do pixel na imagem de entrada for menor ou igual ao *threshold*, então o pixel correspondente na imagem de saída será preto.

Geralmente a binarização da imagem ocorre no seu pré-processamento, visando promover melhor performance no processamento da imagem quando esta entrar no fluxo do modelo de *Deep Learning*.

No entanto, para a obtenção de uma ótima imagem binária, é necessário encontrar o valor de *threshold*, para tanto são efetuadas muitas tentativas com o objetivo de localizar o melhor valor e obter um resultado satisfatório.

Uma forma automática de resolver esse problema é o conceito desenvolvido por Nobuyuki Otsu (YOUSEF, 2011). No conceito de Otsu é trabalhada a imagem original, com a finalidade de obter seu histograma, para então descobrir o valor de *threshold* T e, na imagem de saída, substituir regiões de pixel onde a saturação é maior que T por branco, e no caso oposto, por preto. A função *threshold()* implementada na biblioteca do OpenCV usa o conceito de Otsu.

É possível entender a sintaxe da função como:

`cv.threshold(src, thresh, maxval, type[, dst])` na qual os parâmetros são: *src* indica a imagem original, *thresh* o valor de *threshold*, *maxval* o máximo valor usado para tipos de *threshold* como `THRESH_BINARY` e `THRESH_BINARY_INV`, *type* aponta o tipo de *threshold* adotado e *dst* é um array de saída com o mesmo número, tipo e tamanho de canais da imagem de entrada. A seguir podemos demonstrar o processo para obter a binarização da figura 4.1.6.

Figura 4.1.6 - Imagem original sem aplicação de filtro



Fonte: Google, 2021

O primeiro passo é a conversão da imagem para gradiente cinza, como é visto na figura 28, em seguida obtemos o histograma da imagem em gradientes de cinza, representado no gráfico (Fig. 4.1.7 e Fig. 4.1.8).

Figura 4.1.7 - Imagem pré-processada usando filtros de conversão de cores para cores de cinza



Fonte: Autor, 2021

Figura 4.1.8 - Histograma da imagem após a aplicação do filtro de conversão de cores para cores de cinza



Fonte: Autor, 2021

Após essa preparação, aplica-se a função nativa da biblioteca do OpenCV para binarizar a imagem. A aplicação da função (Fig.4.1.9) é vista com seu respectivo resultado na figura 4.1.10 e seu histograma no gráfico (Fig. 4.1.11).

Figura 4.1.9 - Diagrama da função de binarização de uma imagem



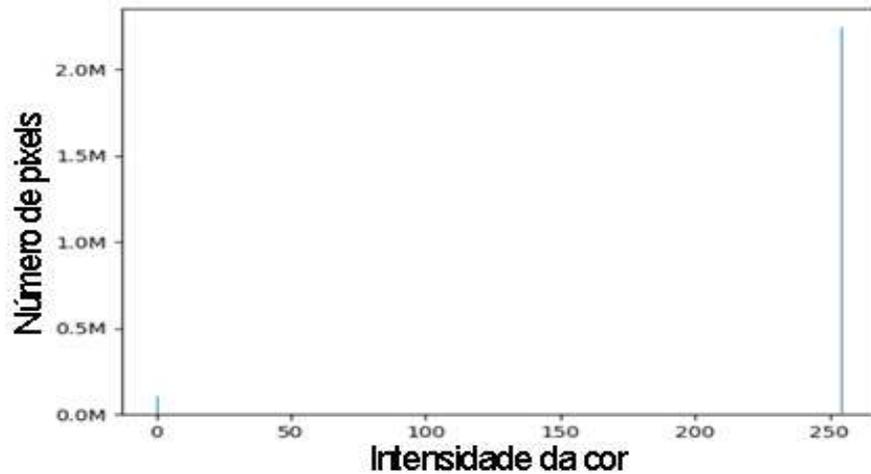
Fonte: Autor

Figura 4.1.10 - Imagem após aplicação da binarização da mesma



Fonte: Autor, 2021

Figura 4.1.11 - Histograma da imagem com o filtro da binarização aplicado

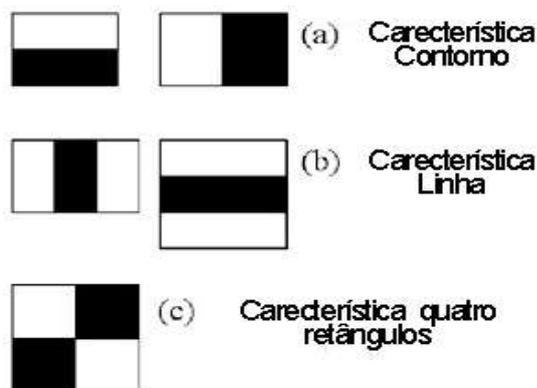


Fonte: Autor, 2021

4.2 CLASSIFICADOR HAAR CASCADE

A detecção de objetos usando um classificador pode ser atingida a partir do princípio do *Haar Cascade*. Esse algoritmo foi proposto por Paul Viola e Michael Jones em seu artigo (VIOLA e JONES, 2001). Nele as características de um objeto podem ser descritas em pequenas partes ou “Haars” (Fig.4.2.1). Dessa maneira, cria-se um classificador que compare a imagem em análise com o classificador treinado, usando essas mesmas características.

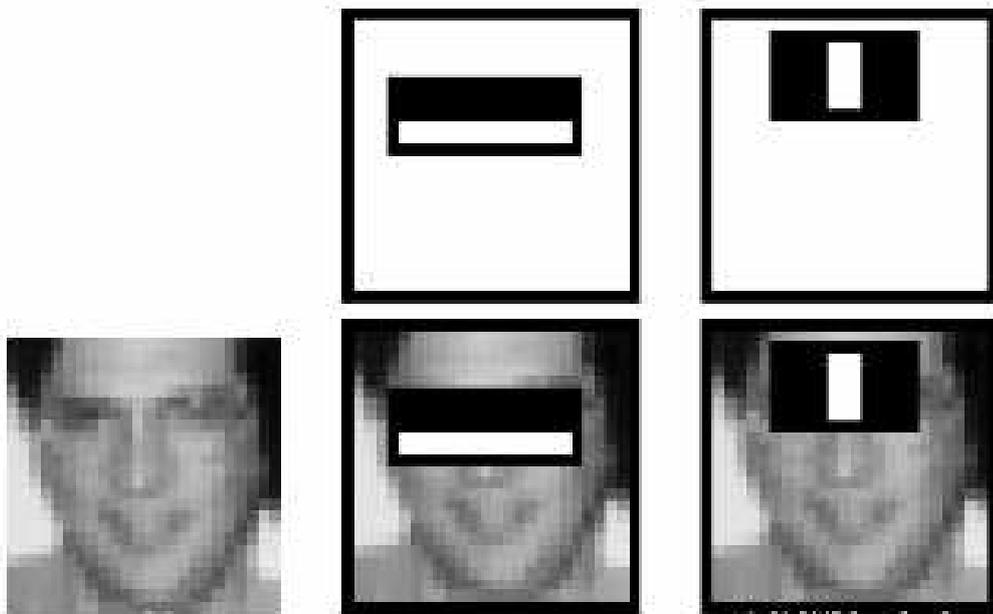
Figura 4.2.1 - Definições de Haars



Fonte: OpenCV 2011

Uma aplicação habitual do classificador *Haar Cascade* seria na detecção de rostos humanos (Fig.4.2.2). A seguir há a imagem de um rosto humano servindo-se da classificação acima citada.

Figura 4.2.2 - Aplicação de Haar Cascade em detecção de rosto humano



Fonte: *OpenCV 2011*

4.3 CANNY DETECÇÃO DE LADOS

A principal característica de um SDC consiste no reconhecimento de ruas durante sua trajetória. Esse algoritmo é muito usado para esse fim, ele diz respeito a como a vizinhança de pixel se comporta em relação a outras vizinhanças. A fórmula que define como se comporta o lado é dado por:

Equação 4.3.1 - Fórmula para detecção de lado usando Canny

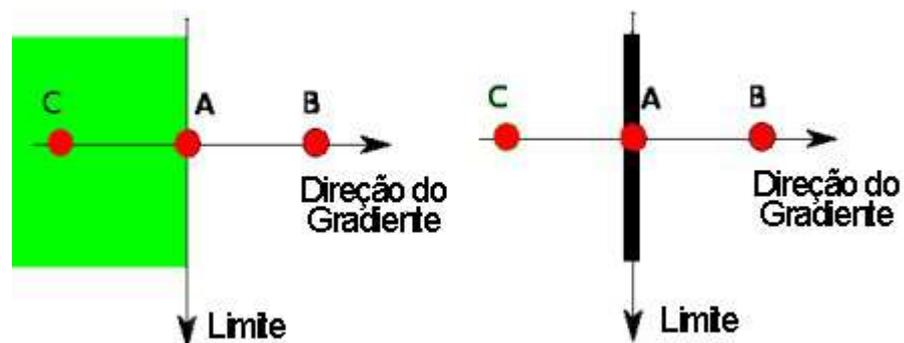
$$\text{Edge_Gradiente } (G) = \sqrt{G_x^2 + G_y^2}$$

$$\hat{\text{Ângulo}} (\Theta) = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

Fonte: *OpenCV 2011*

Gx é a direção horizontal e Gy a direção vertical da imagem. A sintaxe do código no opencv é dada como `cv2.Canny(image, low_threshold, high_threshold)`. Com essa função é possível isolar lados fazendo com que a imagem, depois de convertida em tons de cinza e feito tratamento para amenizar o ruído em sua composição. A representação do uso geral dessa função é dada pela figura a seguir:

Figura 4.13 - Representação geral geométrica do Canny

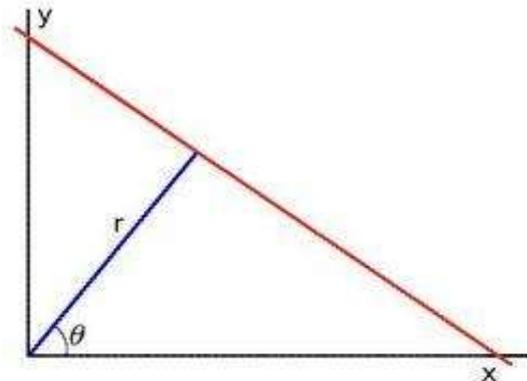


Fonte: *OpenCv 2011*

4.4 TRANSFORMADA DE LINHA DE HOUGH

Esse algoritmo é regularmente usado para detecção de linhas retas em imagens. A base de desenvolvimento é feita na teoria geométrica cartesiana e a relação de pontos e retas. Em um espaço qualquer uma linha pode ser descrita com a sua forma geral $y = mx + b$. Na qual os parâmetros m (coeficiente angular da reta) e b (coeficiente linear da reta) podem ser representados por pontos no espaço da imagem. A seguir temos uma representação geométrica da reta (DUDA e HART, 1972):

Figura 4.14 - Representação geométrica da reta.



Fonte: *OpenCV 2011*

A representação cartesiana pode ser dada como (m, b) e a representação polar como (r, q). Na transformada de *Hough* a representação da reta no sistema polar é descrita como:

Equação 4.4.1 - Representação da eq. da reta no sistema polar

$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right)$$

Fonte: *OpenCV 2011*

Na biblioteca *OpenCV*, essa função possui a seguinte sintaxe: *HoughLines*(image, rho, theta, threshold[, lines[, srn[, stn[, min_theta[, max_theta]]]]]). Para a obtenção de um resultado satisfatório, é necessário um pré-processamento antes do uso dessa função. A melhor forma de preparar a imagem antes da aplicação seria com o uso do algoritmo de detecção de bordas ou *edge detection*.

5 EM BUSCA DA INTELIGÊNCIA ARTIFICIAL

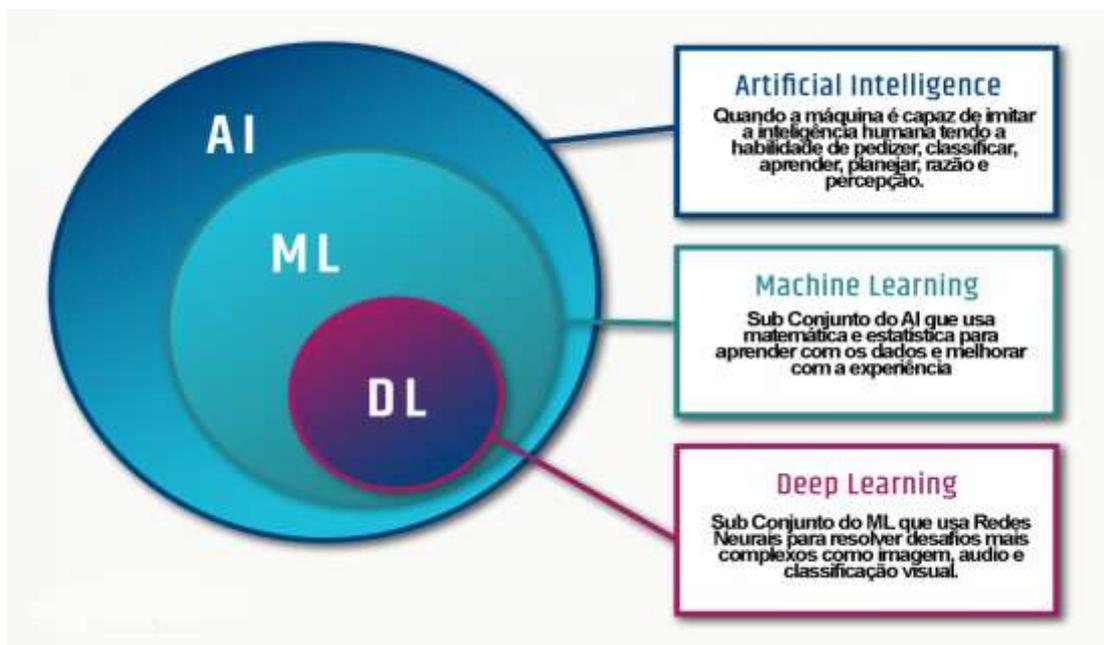
Um grande avanço da robótica está relacionado à Inteligência Artificial. Esse campo da ciência estuda, exclusivamente, como os seres humanos aprendem e busca desenvolver e empregar máquinas que possam realizar atividades humanas de maneira autônoma.

Os sistemas dotados de Inteligência Artificial conseguem executar tarefas que até então eram exclusivamente humanas. Várias aplicações podem ser mencionadas, dentre as mais comuns estão: reconhecimento de voz, conversação com seres humanos e reconhecimento visual.

A Inteligência Artificial (IA) começou a ser estudada na década de 50 quando pesquisadores começaram a ter uma compreensão mais sólida de como funcionavam os neurônios humanos. A partir de então, conseguiram desenvolver modelos matemáticos de redes neurais e reproduziram em forma de algoritmos.

As subdivisões do IA podem ser representadas como *Machine Learning (ML)* e *Deep Learning (DL)* (Fig.4.4.1). Ambas usam princípios de redes neurais e aprendizagem cognitiva humana. A seguir temos a representação visual das divisões que compõem a Inteligência Artificial:

Figura 4.4.1 - Subdivisões da Inteligência Artificial



Fonte: Analytics 2020

O *Machine Learning* ou ML é uma subdivisão da Inteligência Artificial que usa conceitos de big data e aprendizagem cognitiva para extrair melhores resultados e, dessa forma, alcançar maior adaptação ao meio em que está inserido. O *Deep Learning* é outra subdivisão do *Machine Learning* que usa conceitos de Redes Neurais e como elas lidam com informações.

5.1 DEEP LEARNING

No artigo de 2006, Geoffrey Hinton é um dos pioneiros no campo das Redes Neurais Artificiais. Ele estabeleceu a necessidade de usar muitas “camadas” para se obter um aprendizado mais eficiente, é sua a expressão “*Deep Learning*” (HINTON, OSINDERO e TEH, 2006). O modelo mais utilizado é *Convolutional Neural Networks* (CNNs), a ideia principal desse modelo é baseada no artigo de KUNIHICO FUKUSHIMA (FUKUSHIMA, 1988), no qual desenvolve-se um modelo de várias camadas de redes neurais capazes de aprender características de imagens e realizar várias tarefas como: classificação de objetos, detecção e segmentação.

As arquiteturas do DL são classificadas em três categorias. A primeira aborda a aprendizagem supervisionada, quando o algoritmo sabe os dados de entrada e os de saída. A segunda pertence à aprendizagem não supervisionada, quando o algoritmo sabe os dados de entrada e tem que descobrir os de saída. E a terceira refere-se à Aprendizagem por Reforço ou *Reinforcement Learning*, que consiste em um treinamento de modelos com o propósito de tomar sequência de decisões.

A máquina aprende a atingir uma certa meta em um ambiente incerto e complexo. O modelo de DL utiliza tentativa e erro para encontrar a solução do problema. A programação é baseada em recompensas e penalidades para cada ação (SUTTON e BARTO, 2018).

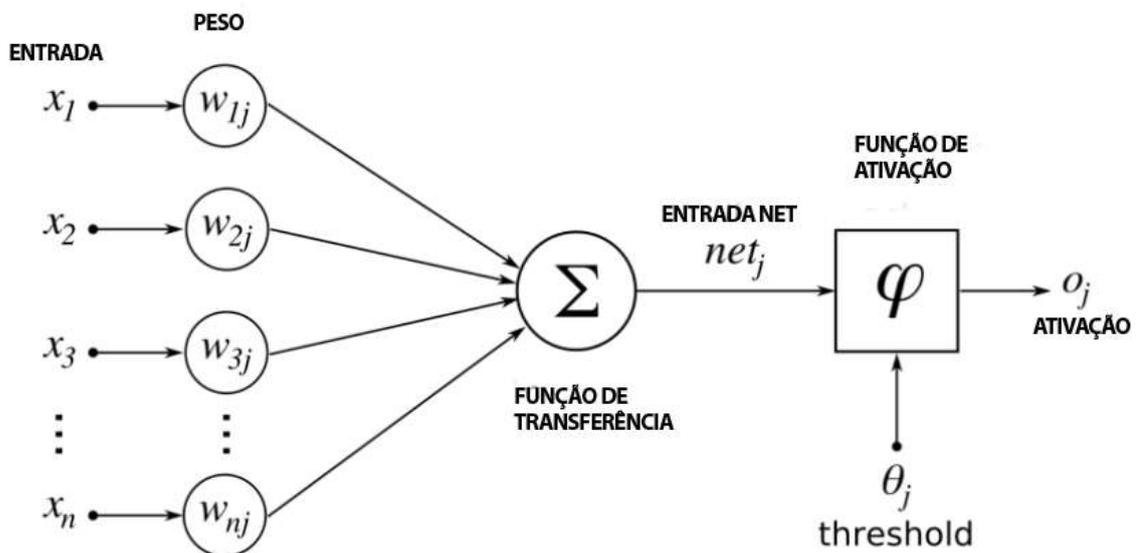
5.2 CONVOLUTIONAL NEURAL NETWORKS

Em 1980, Yann LeCun, um estudante de doutorado, apresentou um artigo sobre Convolutional Neural Network (Yann LeCun Interview), ou mais conhecido como ConvNets. LeCun usou como base para seu estudo o trabalho do pesquisador japonês Kunihiko Fukushima, que anos antes apresentou uma forma rudimentar de reconhecimento de imagem por neural network chamado *neocognitron*. Suas primeiras aplicações se deram no reconhecimento de letra manuscrita, e logo foi

adotada em correios e bancos. Nos correios, esse algoritmo era treinado para reconhecer CEP e agilizar a separação de cartas e pacotes.

CNN é um conjunto de vários *neurons*. *Neurons* (Fig. 5.2.1) é a representação matemática de neurônios biológicos, para isso podemos entender a representação matemática de um neurônio.

Figura 5.2.1 - Estrutura matemática de funcionamento de um *neuron*



Fonte: Autor

O comportamento de cada *neuron* é definido pelo peso (forças de conexão sináptica) que foi estipulado. Quando uma imagem é inserida no ConvNet, cada camada gera vários mapas de ativação. Os mapas de ativação são características realçadas. O somatório de todas as entradas x_n multiplicada por seus respectivos pesos resulta em net de um *neuron*. Essa operação é chamada de convolução, daí a origem do nome Convolutional Neural Network (Eq.5.2.1).

Equação 5.2.1 - Equação de um neurônio artificial

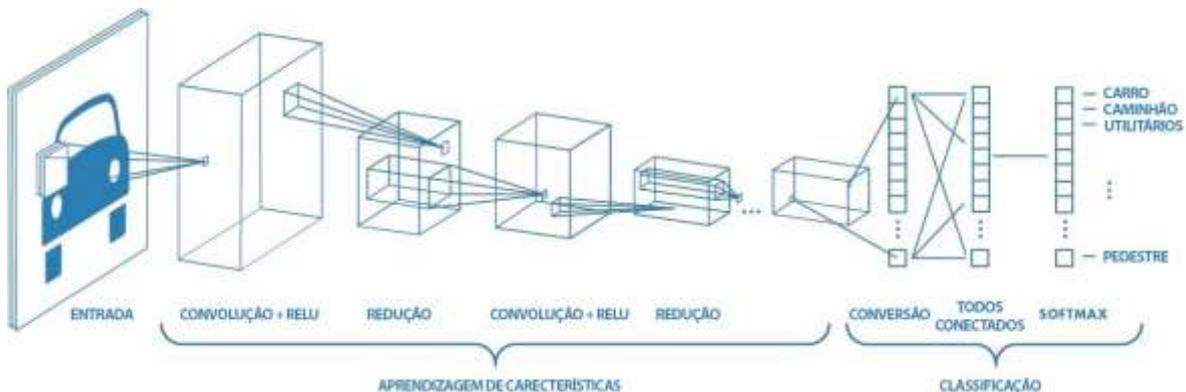
$$net_i(t) = \sum_{j=1}^n w_{ij}x_i(t)$$

Fonte: Autor

Após a determinação do net, a função de ativação atualiza o valor de ativação. A função de ativação é dada pela equação: $x(t+1) = fa(x(t), net(t))$. Os estados futuros do *neuron* são influenciados pelo seu estado atual e pelo valor de entrada net. A primeira camada do CNN geralmente detecta características básicas da imagem, como por exemplo, contornos horizontais, verticais e diagonais. A saída para a primeira é alimentada na camada seguinte, na qual são extraídas características mais complexas em comparação à camada anterior. Conforme o algoritmo vai se aprofundando no Convolutional Neural Network, as camadas conseguem detectar objetos mais complexos como rostos e objetos específicos.

A classificação da última camada se baseia no mapa de ativação obtido na última camada de convolução, gerando um conjunto de estatísticas de confiança que varia de 0 a 1 e especifica a que classe a imagem pertence. Um exemplo de ConvNet usado para detectar veículos e pedestres pode ser melhor entendido na figura 5.2.2.

Figura 5.2.2 - Modelo simplificado do funcionamento de uma CNN



Fonte: <https://towardsdatascience.com/>

Uma das partes mais difíceis de desenvolver CNN é ajustar os pesos individualmente em cada neurônio para extrair corretamente as características da imagem. Esse processo de ajustar o peso corretamente é chamado de treinamento do CNN para que a cada interação ele possa ajustar os pesos de cada neurônio. No

começo, o modelo do CNN tem números randômicos dos pesos. No treinamento, o algoritmo usa um grande *dataset* com a classe conhecida, como por exemplo carros, gatos, etc.

No começo, o ConvNet processa as imagens com números randômicos no peso em cada neurônio, a seguir compara com a imagem de saída e o resultado obtido. Caso não coincida o com o nome da imagem, o ConvNet ajusta os pesos para que a interação seguinte resulte o mais próximo possível da resposta correta.

O nome desse processo de correção é chamado *backpropagation* ou *backprop*. Esse sistema otimiza o processo de ajuste do parâmetro e facilita encontrar o parâmetro que deve ser ajustado. Cada interação é chamada de *epoch*.

O ConvNet passa por vários *epochs* durante o treinamento, fazendo pequenos ajustes no parâmetro do peso. A cada *epoch*, os neurônios artificiais se tornam cada vez melhores para classificar imagens, se convergem em um determinado ponto quando tornam-se bons em classificar determinada imagem a qual foram treinados.

Após o treinamento, usa-se um *dataset*, no qual é conhecida a saída das imagens ou as classes às quais as imagens pertencem, com o objetivo de testar a precisão do modelo em classificar as imagens. Vale lembrar que o *dataset* usado para fazer o teste não é o mesmo para fazer o treinamento, sendo, portanto, imagens inéditas para o ConvNet.

Pode acontecer de o modelo ter uma boa performance no treinamento, mas no teste apresentar uma baixa performance. Isso acontece devido “*overfitted*”, ou seja, no processo de treinamento não houve uma variedade suficiente no *dataset* das imagens para alcançar um modelo robusto.

Em muitos casos não há necessidade de obter as imagens para treinar o ConvNet a cada vez, é possível usar banco de *datasets* com imagens já classificadas. Um exemplo desses *datasets* que estão disponíveis é o MNIST com um banco de mais de 70.000 imagens.

5.3 APRENDIZAGEM POR REFORÇO (RL)

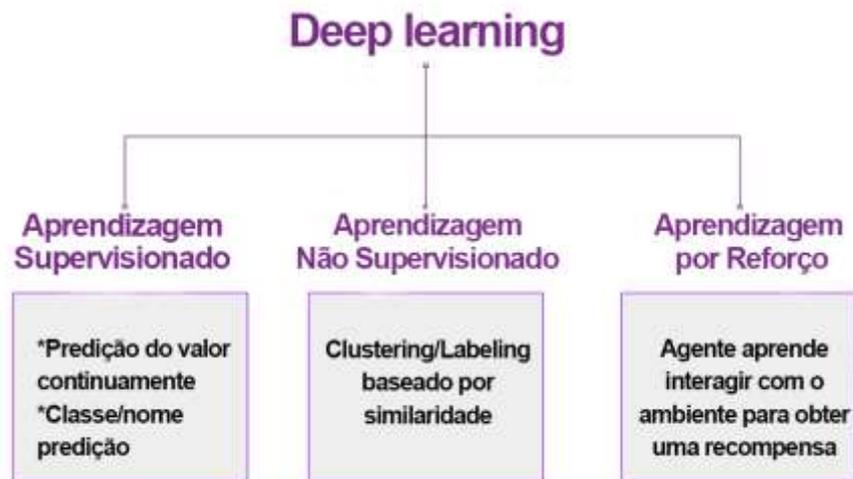
Aprendizagem por reforço é o termo formal em que o modelo de *Deep Learning* aprende a atuar em certos ambientes que levam a uma recompensa máxima. Esse método consiste basicamente na exploração do ambiente para acumular recompensas. Os modelos de aprendizagem podem ser vistos na figura 5.3.1.

No modelo de aprendizagem supervisionado, o exemplar aprende sob supervisão. Neste caso, aprende com um *dataset* com classe e nome conhecidos e quando uma nova imagem aparece, o modelo a compara com as informações conhecidas para fazer um confronto e chegar em uma predição do resultado.

Existem duas grandes áreas onde o modelo de aprendizagem supervisionada tem uma ótima aplicação, estas são: problemas de classificação e problemas de regressão (Fig.5.3.1).

Ao contrário do modelo de aprendizagem supervisionado, o modelo de aprendizagem não supervisionado usa *dataset* que não tem informações detalhadas ou instruções específicas. O *dataset* de treinamento não tem nenhuma informação ou contém poucas informações para o modelo, proporcionando ao algoritmo a tentativa de encontrar automaticamente a estrutura dos dados pela análise e extração de características. Dependendo do *dataset*, o algoritmo pode organizar as seguintes formas: *clustering*, detecção de anomalias, associação e *autoencoders*. Nesse modelo é muito difícil medir sua precisão.

Figura 5.3.1 - Métodos de aprendizagem no *Deep Learning*



Fonte: Autor

Pode se fazer uma comparação a um videogame, quando um nível é completado conseguimos um troféu, se vencemos um chefe conseguimos bônus de pontos e se pisamos em uma armadilha morremos. Com essas informações, o jogador consegue melhorar na próxima partida, sem essas informações o jogador levaria muito mais tempo tomando ações aleatórias esperando prosseguir para o próximo nível. Os agentes são encorajados a achar um caminho para conseguir um determinado objetivo ou melhorar o desempenho em uma determinada tarefa. Conforme o agente vai progredindo e fica mais perto de alcançar o objetivo, ele recebe recompensas.

Para conseguir tomar decisões, o agente possui o conhecimento das interações passadas e a exploração que promovem recompensas maiores. O agente tenta aumentar as recompensas acumulativas. Esse modelo é um processo iterativo, no qual a cada interação com *feedbacks*, melhor a estratégia do agente se torna. Esse processo é muito usado para treinar robôs como por exemplo decisões de mudar a direção de carros autônomos ou controlar eficientemente o estoque de uma fábrica.

Antes de prosseguirmos, vamos definir a terminologia básica do modelo de aprendizagem por reforço:

- Agente: Objeto ou sujeito que executa ações em um determinado ambiente para ganhar recompensa.
- Ação: Todos os possíveis movimentos que o agente pode executar.
- Ambiente: o cenário onde o agente tem que enfrentar.
- Estado (s): Situação atual do agente que o ambiente retorna.
- Recompensa (R): Retorno imediato enviado do ambiente após uma ação no estado atual.
- Política (π): A estratégia que o agente aplica para achar a próxima ação baseada no estado atual.
- Valor (V): é o retorno a longo prazo, ao contrário da recompensa que é imediata (R). $V\pi(s)$ é definido como o retorno esperado a longo prazo do estado atual (s) afetado pela política (π).
- Q-Valor ou Valor-ação (Q): O conceito do Q-Valor é similar ao Valor, apenas com o parâmetro a mais, a ação atual (a). $Q\pi(s, a)$ refere-se ao

retorno de longo prazo do estado atual (s), tomando uma ação (a) afetado pela política (π).

Existe três formas de se implementar o modelo de aprendizagem por reforço:

- Valor-Baseado: nesse método, tenta-se maximizar o valor da função $V(s)$. Como foi definido anteriormente, $V_\pi(s)$ é definido como o retorno esperado a longo prazo do estado atual (s) afetado pela política (π). Com isso, $V(s)$ é o valor da recompensa que o agente espera obter no futuro, representada pela equação:

Equação 5.3.1 - Equação matemática do método de aprendizagem por reforço baseado no valor

$$V_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s]$$

Fonte: <https://wiki.pathmind.com/deep-reinforcement-learning>

- Política-Baseado: Nesse método tenta-se obter uma política no qual a ação executada em cada estado é otimizada para ganhar a máxima recompensa no futuro. Sabendo disso, sabemos que a política (π) determina a próxima ação (a) em qualquer estado (s). Existe dois tipos de métodos de política-baseado:
 - Determinístico: Em qualquer estado (s), a mesma ação (a) é produzido pela política (π).
 - Estocástico: A cada ação tem uma certa probabilidade dada pela equação:

Equação 5.3.2 - Equação estocástica do modelo da política-baseado

$$\pi(a|s) = P[A_t = a | S_t = s]$$

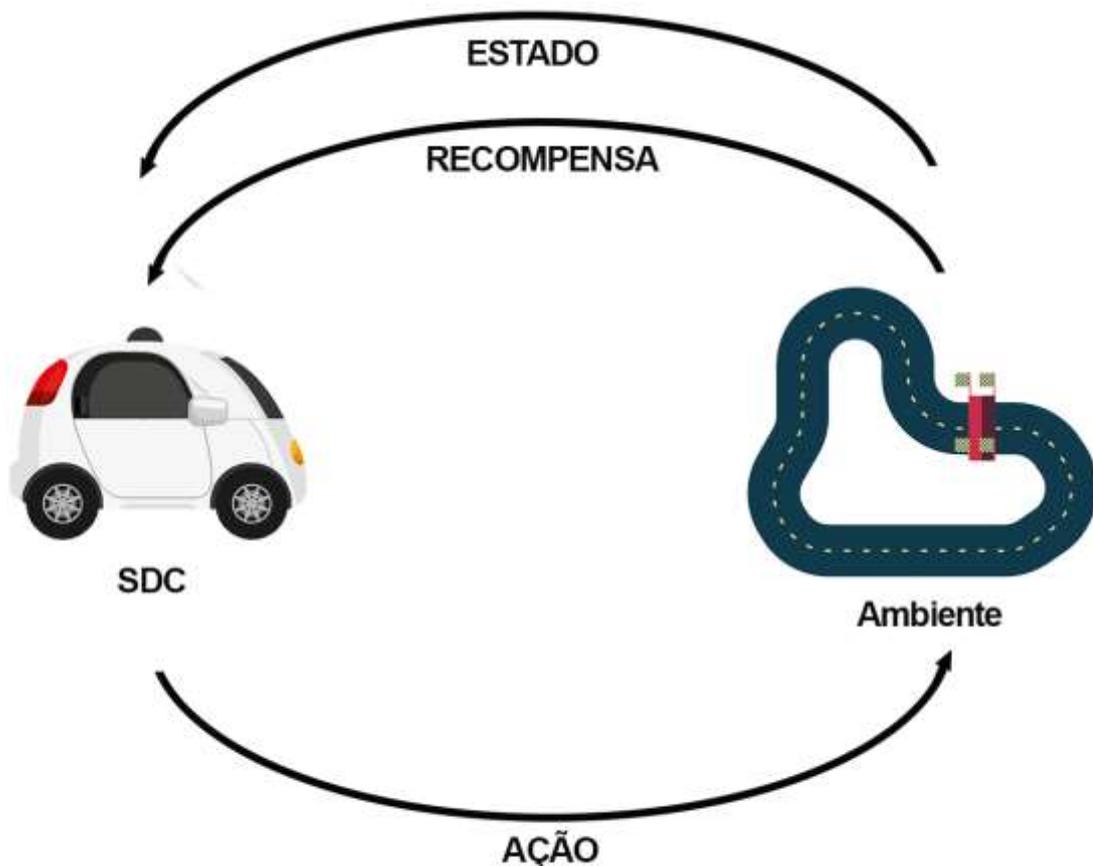
Fonte: <https://wiki.pathmind.com/deep-reinforcement-learning>

- Modelo-Baseado: Nesse modelo é criado um protótipo virtual para diferentes ambientes, este aprende, por sua vez, a se comportar em cada um deles. Como o modelo difere a cada ambiente, não há um algoritmo ou uma solução singular para ele.

5.4 APLICAÇÃO DA APRENDIZAGEM POR REFORÇO NO PROJETO

Em nosso projeto, o carro usa o modelo RL para explorar o ambiente em que está inserido, executando tarefas com respostas positivas e evitando respostas negativas. O objetivo do modelo é conseguir o máximo possível de respostas positivas a partir da interação com o meio. A representação básica do funcionamento do modelo RL pode ser vista na Figura 5.4.1.

Figura 5.4.1 - Loop de funcionamento do Modelo

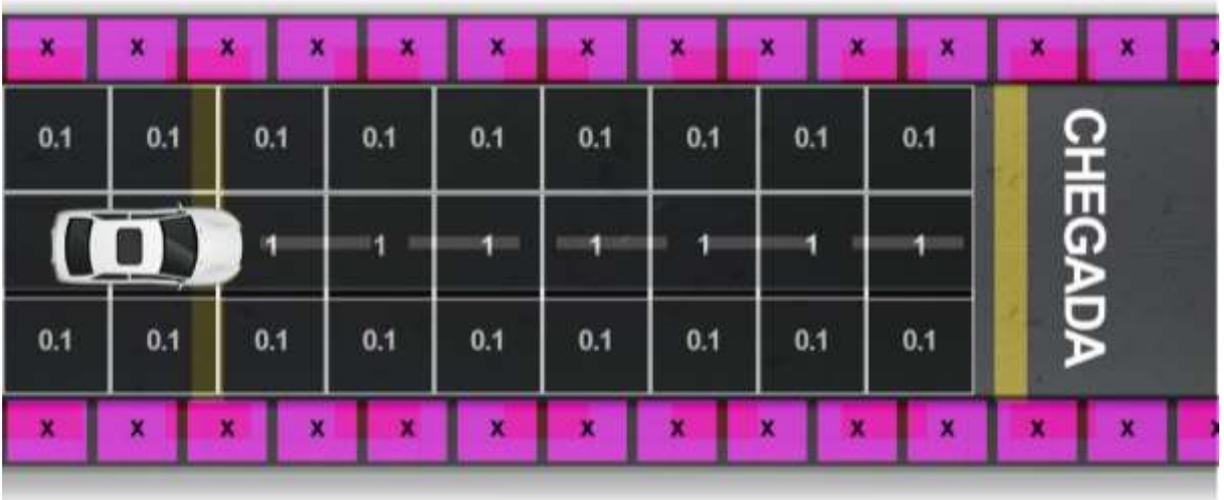


Fonte: Autor

A ação que o SDC executa é a correção do ângulo da direção e a velocidade do carro conforme o dado de entrada. A parte do estado é o dado de entrada captado pela câmera frontal do SDC iniciando assim uma nova ação. Para podermos aplicar o RL ao modelo, podemos dividir a trajetória e atribuir pontuação para cada ponto do

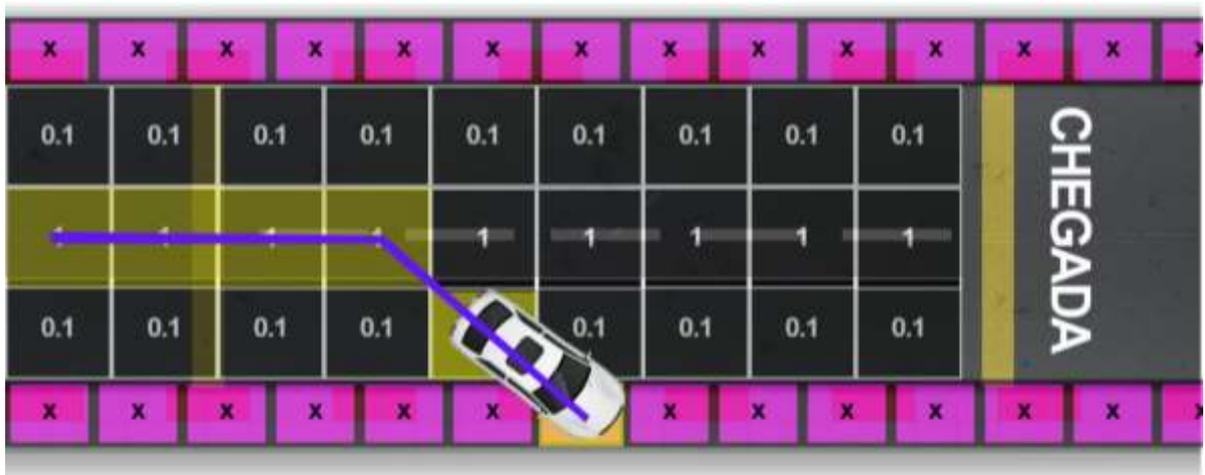
grid da trajetória (Fig. 5.4.2), podemos colocar delimitadores (X) na trajetória para que o nosso SDC saiba que está fora do percurso. No treinamento, o SDC inicia a exploração e a cada interação acumula experiência. O modelo aprende a cada interação e, dessa forma, conseguindo mais pontos até alcançar o objetivo (Fig. 5.4.3).

Figura 5.4.2 - Trajetória dividida em Grid e com pontos atribuídos



Fonte: Autor

Figura 5.4.3 -Interação acumulando pontos



Fonte: Autor

6 METODOLOGIA DE PESQUISA

Para a resolução do problema proposto neste trabalho foi usada a metodologia de pesquisa experimental. Foram feitas observações a partir de simulações controladas, com alterações de parâmetros no modelo desenvolvido a ser testado na simulação. Além disso, técnicas de análise de dados quantitativas e qualitativas foram empregadas nas observações do comportamento durante o ensaio.

A linha de pesquisa de SDC é bastante dispendiosa, seja no meio acadêmico ou mesmo para grandes empresas automobilísticas, por isso é comum o uso de simuladores. Nestes são recriadas todas as condições do meio real. Atualmente existem vários tipos de simuladores (Anexo IV) que conseguem reproduzir com grande precisão o mundo real. No entanto, foi escolhido o simulador da AWS, no qual podem ser usados os servidores da AWS para efetuar a simulação, facilitando o desenvolvimento e o teste.

6.1 AS FERRAMENTAS

Para conseguirmos desenvolver esse trabalho foram usadas algumas ferramentas descritas neste capítulo. Quase todas as ferramentas apresentadas são *open-sources*, por isso observam-se algumas limitações, contudo foram obtidos bons resultados.

A base da programação foi feita na linguagem python, uma linguagem muito comum no manuseio de grandes arquivos em inteligência artificial. Além disso, foram usados vários frameworks, como o *TensorFlow 2.3.0* e *Keras*.

O Google Colab auxiliou o desenvolvimento do modelo de Inteligência Artificial. No capítulo 8 é explicado de forma mais detalhada o uso do Colab no desenvolvimento do algoritmo.

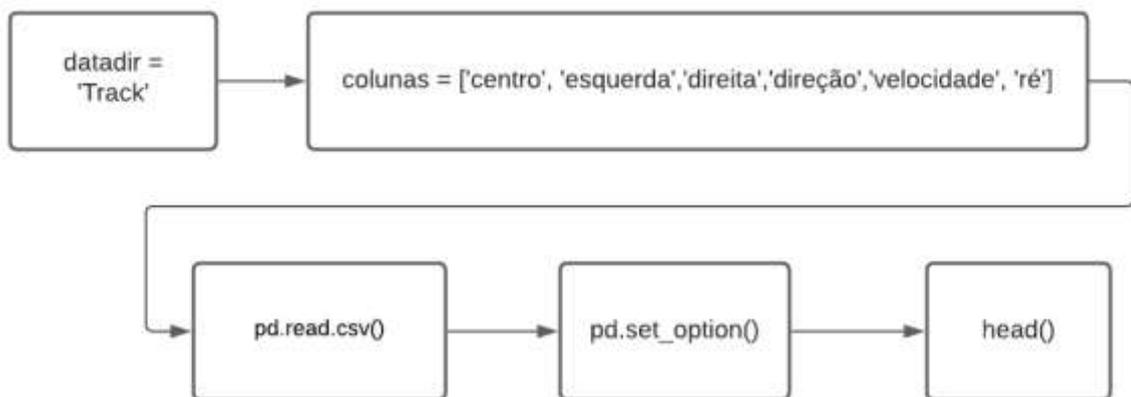
Usamos também o ambiente de simulação da AWS, nele foi definida a arquitetura CNN na qual se baseia o modelo RL desenvolvido. O ambiente de simulação da AWS é um serviço pago conforme uso. Várias versões foram preparadas, primeiramente testamos no IDE do Google Colab e depois comprovamos

sua validação no ambiente da AWS. Por ser um serviço pago, as versões foram restringidas para as que apresentaram os melhores resultados.

7 DESENVOLVIMENTO DO ALGORITMO DO MODELO

O primeiro passo foi obter um *dataset* para treinar o IA. Para isso, o recurso do simulador foi utilizado para a obtenção das imagens e seus principais dados, como velocidade e ângulo da direção. Uma câmera no simulador auxiliou a gravação das imagens (Fig. 6.1.1 e Apêndice A), gerou um arquivo csv com a descrição da imagem, o ângulo da direção, no momento de gravação, bem como a velocidade.

Figura 6.1.1 - Diagrama da criação de *dataset*



Fonte: Autor

Figura 6.1.2 - Configuração do SDC com uma câmera frontal



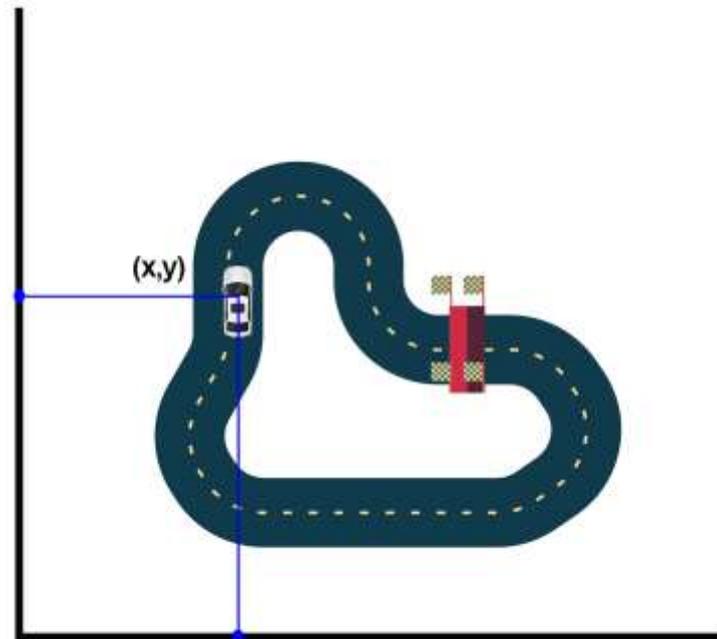
Fonte: AWS

O desenvolvimento do modelo RL é feito para que o SDC (Fig.6.1.2) explore o ambiente e acumule recompensas ao longo das interações. Dessa maneira o modelo aprende a cada ação que resulta em altas recompensas, aproximando-se do objetivo principal. Podemos usar alguns parâmetros para reforçar a recompensa no algoritmo do modelo. A seguir vamos detalhar cada um desses parâmetros.

7.1 PARÂMETRO DE POSICIONAMENTO NA TRAJETÓRIA

O parâmetro reproduz o posicionamento geográfico do SDC. Na simulação ele é a representação cartesiana em um determinado tempo na trajetória (Fig.7.1.1).

Figura 7.1.1 - Parâmetro de posicionamento na trajetória



Fonte: Autor

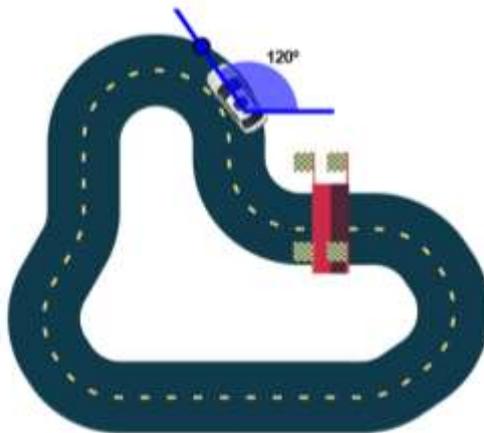
Algo que está sendo bastante considerado em estudos avançados de *smart roads*, é a conexão de rede 5G, principalmente na comunicação entre veículos, ou *Vehicle-to-vehicle* (V2V) (DEY, RAYAMAJHI, *et al.*, 2016). Um consórcio de aproximadamente 120 membros, está desenvolvendo a tecnologia para conexão 5GAA em carros autônomos. Uniram-se, neste sentido, grandes empresas que destinam cada vez mais recursos em pesquisas em mobilidade autônoma como: Audi, BMW, Honda, Intel, Samsung e Bosch. Como resultado direto, podemos exemplificar o modelo de conexão C-V2X que não só engloba o V2V, como também define a *vehicle to infrastructure* (V2I). Esta seria a forma que o SDC se comunicaria com a *smart road*. O *vehicle to pedestrian* (V2P) seria a forma como o SDC se comunicaria

com pedestres e usuários. A Letra C desse modelo mostra que a conexão ocorre na mesma rede dos nossos celulares (CHEN, HU, *et al.*, 2002).

7.2 PARÂMETRO DE HEADING DO SDC

Nesse parâmetro, define-se a orientação do SDC em um determinado tempo. Para facilitar a codificação usa-se o mesmo sentido do grid do ambiente no qual foi configurado (Fig.7.2.1).

Figura 7.2.1 - Parâmetro de *Heading* do SDC



Fonte: Autor

7.3 PARÂMETRO DA DISTÂNCIA AO CENTRO DA LINHA

Podemos definir o parâmetro da distância do SDC até a linha central da trajetória, podemos também definir um outro parâmetro do tipo booleano que verifica se o SDC está à esquerda da linha central (Fig. 7.3.1).

Figura 7.3.1 - Parâmetro de distância do SDC até a linha central da trajetória

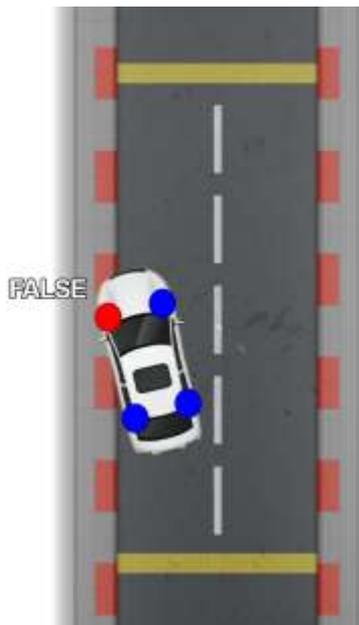


Fonte: Autor

7.4 PARÂMETRO DE TODAS AS RODAS NA TRAJETÓRIA

Esse parâmetro, do tipo booleano, verifica se todas as rodas estão dentro da trajetória. Por definição, se todas as rodas estiverem dentro da trajetória o parâmetro será verdadeiro, caso contrário, será falso (Fig. 7.4.1).

Figura 7.4.1 - Parâmetro que verifica se todas as rodas estão dentro da trajetória

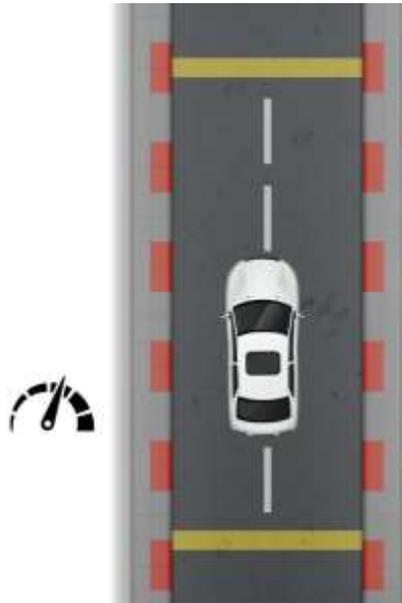


Fonte: Autor

7.5 PARÂMETRO DE VELOCIDADE

Um dos parâmetros mais importantes é o da velocidade que o SDC emprega. Normalmente esse parâmetro é reativo a cada estado que o SDC se encontra (Fig. 7.5.1).

Figura 7.5.1 - Parâmetro de velocidade do SDC

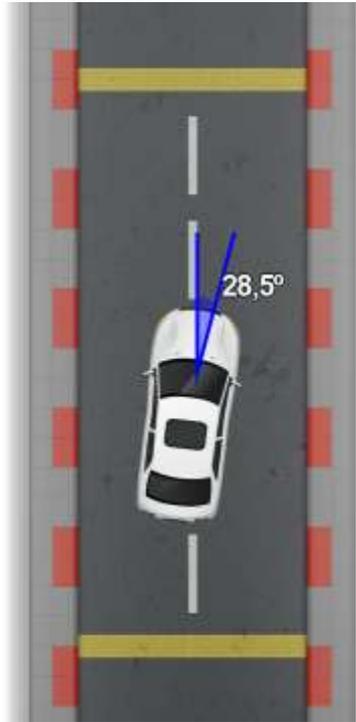


Fonte: Autor

7.6 PARÂMETRO DO ÂNGULO DA DIREÇÃO

O parâmetro do ângulo da direção, conforme o parâmetro de velocidade, também é reativo ao estado anterior do SDC. Ambos são dinâmicos e sofrem alterações, fazendo correções a entrada de cada interação do SDC (Fig. 7.6.1).

Figura 7.6.1 - Parâmetro do ângulo da direção do SDC

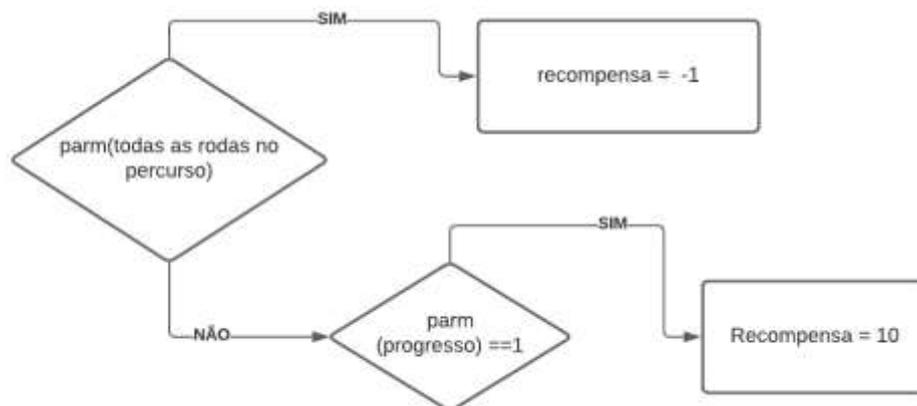


Fonte: Autor

7.7 CODIFICAÇÃO DO ALGORITMO COM OS PARÂMETROS

Os parâmetros listados anteriormente são os mais importantes para a criação do algoritmo do modelo. Desenvolvemos uma função de recompensa, por meio da qual o modelo verifica, a cada interação, como são configurados os parâmetros. Na figura 7.7.1 temos um exemplo da criação da função de recompensa do modelo RL.

Figura 7.7.1 - Diagrama de um exemplo da função recompensa



Fonte: Autor

O desenvolvimento do algoritmo é baseado no artigo do *Proximal Policy Optimization Algorithms* (PPO) (SCHULMAN, WOLSKI, *et al.*, 2017). Neste artigo são propostas novas políticas e métodos de aprendizagem por reforço que utilizam gradiente estocástico para análise dos dados de entrada na sua interação com o ambiente. O simulador da AWS (Fig.7.7.2) disponibiliza esse framework do algoritmo PPO para conseguir um treinamento mais performático.

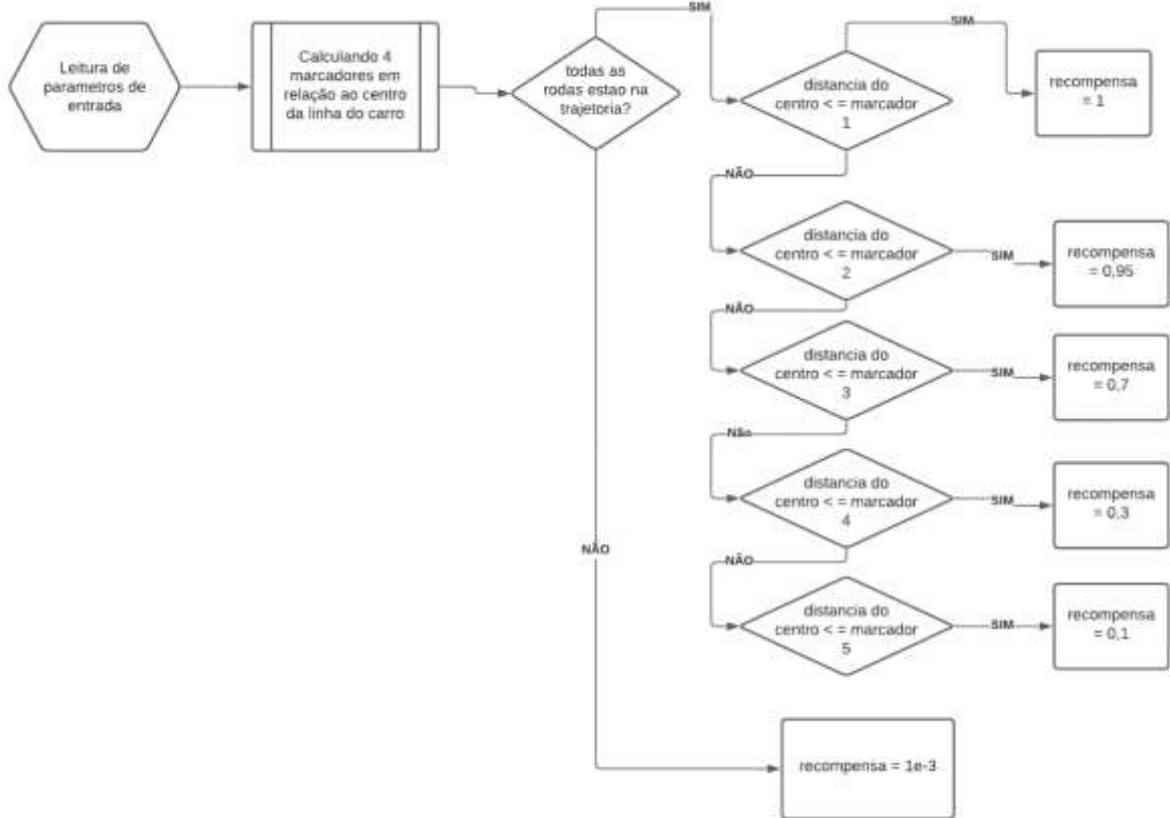
Figura 7.7.2 - Visão da simulação do SDC no AWS



Fonte: Autor

A seguir vamos apresentar as versões usadas para fazer a simulação e, no capítulo seguinte, iremos analisar o desempenho de cada uma delas em comparação às diferentes variantes (Fig.7.7.3).

Figura 7.7.3 - Diagrama da versão do modelo RL



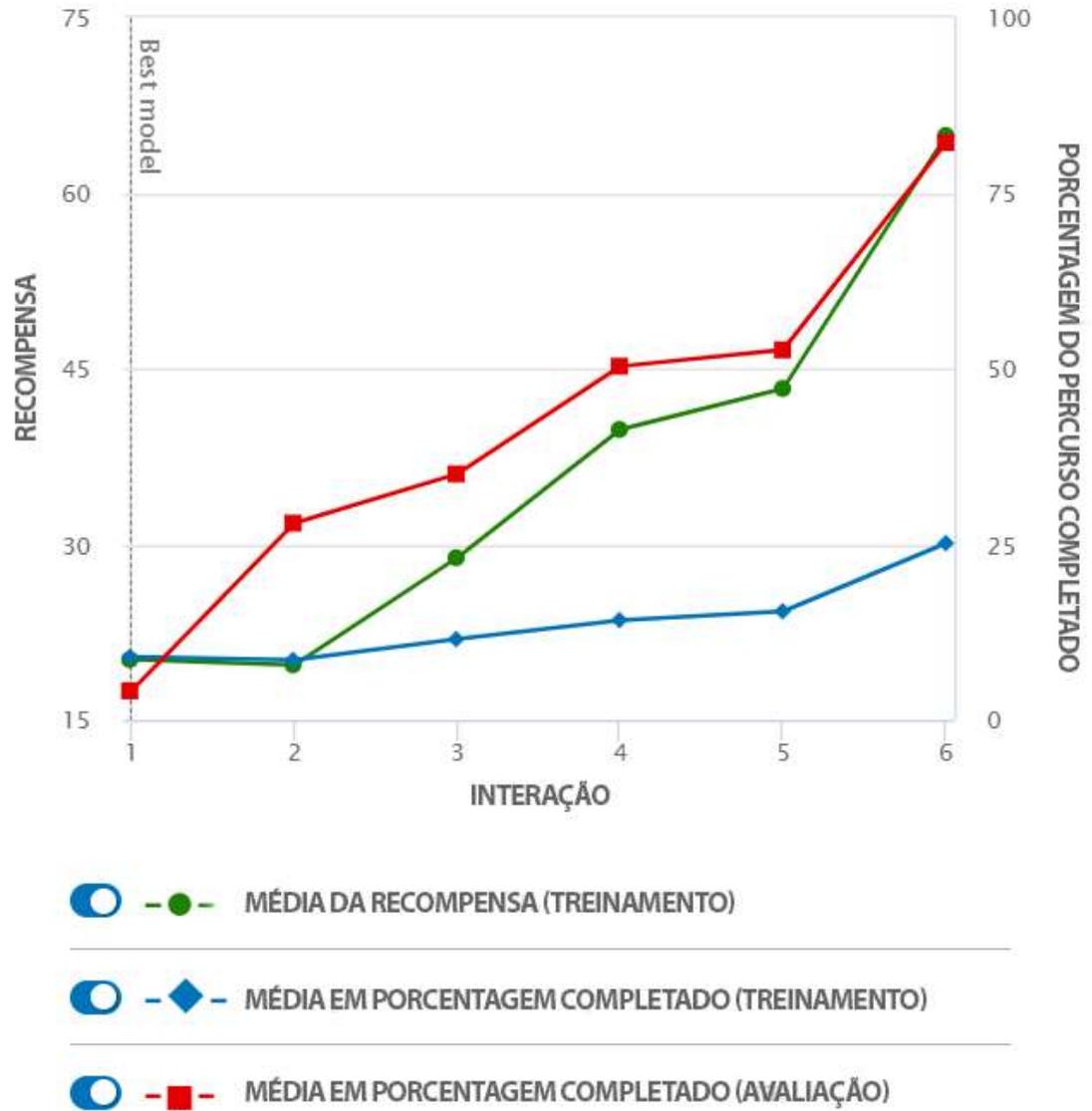
Fonte: Autor

8 ANÁLISE DE DADOS

Na primeira versão do modelo RL (Gráfico 8.1.1), usamos os parâmetros com valores baixos de recompensa, a configuração do CNN foi a mesma para todas as versões e são apresentadas na Tabela 8.1.1.

8.1 VERSÃO 1 DO MODELO RL

Gráfico 8.1.1 - Gráfico do desempenho da versão 1 do modelo RL



Fonte: Autor

Tabela 8.1.1 - Hyperparâmetros do CNN para as versões

Hyperparametro	Valor
Tamanho Gradiente <i>Batch</i> descendente	64
Entropia	0.01
Fator de Desconto	0.999
Tipo de Perda	Hubber
Taxa de aprendizagem	0.0003
Números de episódios de experiencia entre cada interação da Atualização da Política	20
Números de epochs	10

Fonte: AWS

Observou-se que na primeira versão o modelo não conseguiu finalizar a trajetória inteira. Além disso, a massa coletada para o treinamento não foi suficiente para entregar um modelo RL robusto.

No gráfico 8.1.1 (Tab.8.1.2) conseguimos constatar a evolução do modelo e seu comportamento nas tentativas de finalização da trajetória (Fig. 8.1.1). Na avaliação do protótipo, reconhecemos que nas três tentativas, ele conseguiu completar duas trajetórias, em uma delas ele saiu do percurso delimitado.

Tabela 8.1.2 - Evolução da Versão 1 do Modelo RL

Tentativas	Tempo (MM:SS.mmm)	Resultados das Tentativas (% Percurso Completado)	Status
1	00:29.486	100%	Volta Completa
2	00:08.076	27%	Fora do percurso
3	00:29.622	100%	Volta Completa

Fonte: Autor

Figura 8.1.1 - Modelo e descrição da trajetória usada no desenvolvimento das versões do modelo RL

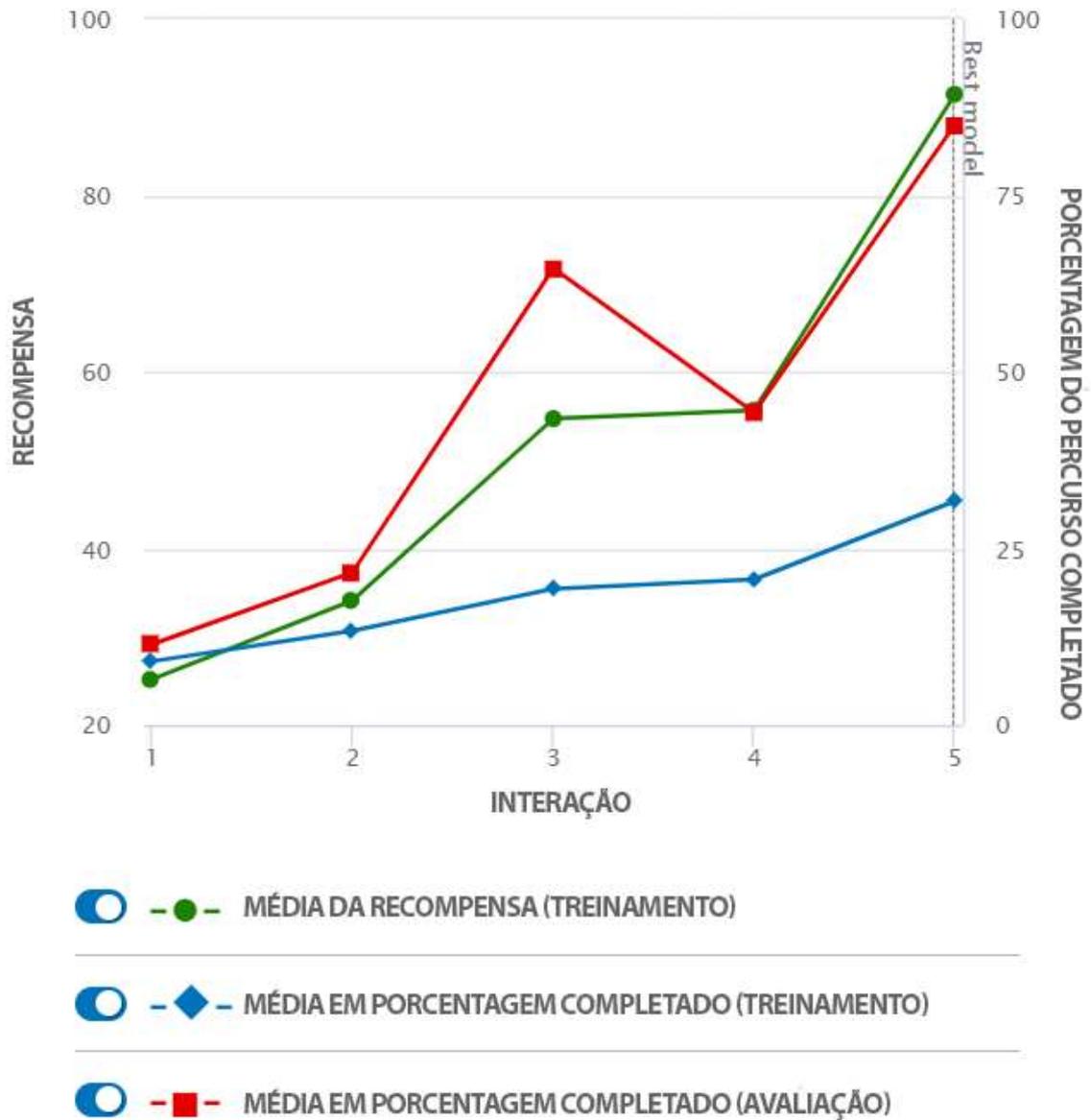


Fonte: AWS

8.2 VERSÃO 2 DO MODELO RL

Na segunda versão (Gráfico 8.2.1) não conseguimos um bom resultado (Tab. 8.2.1), apesar das implementações realizadas nos parâmetros.

Gráfico 8.2.1 - Gráfico da Versão 2 do modelo RL



Fonte: Autor

Tabela 8.2.1 - Evolução da versão 2 do modelo RL

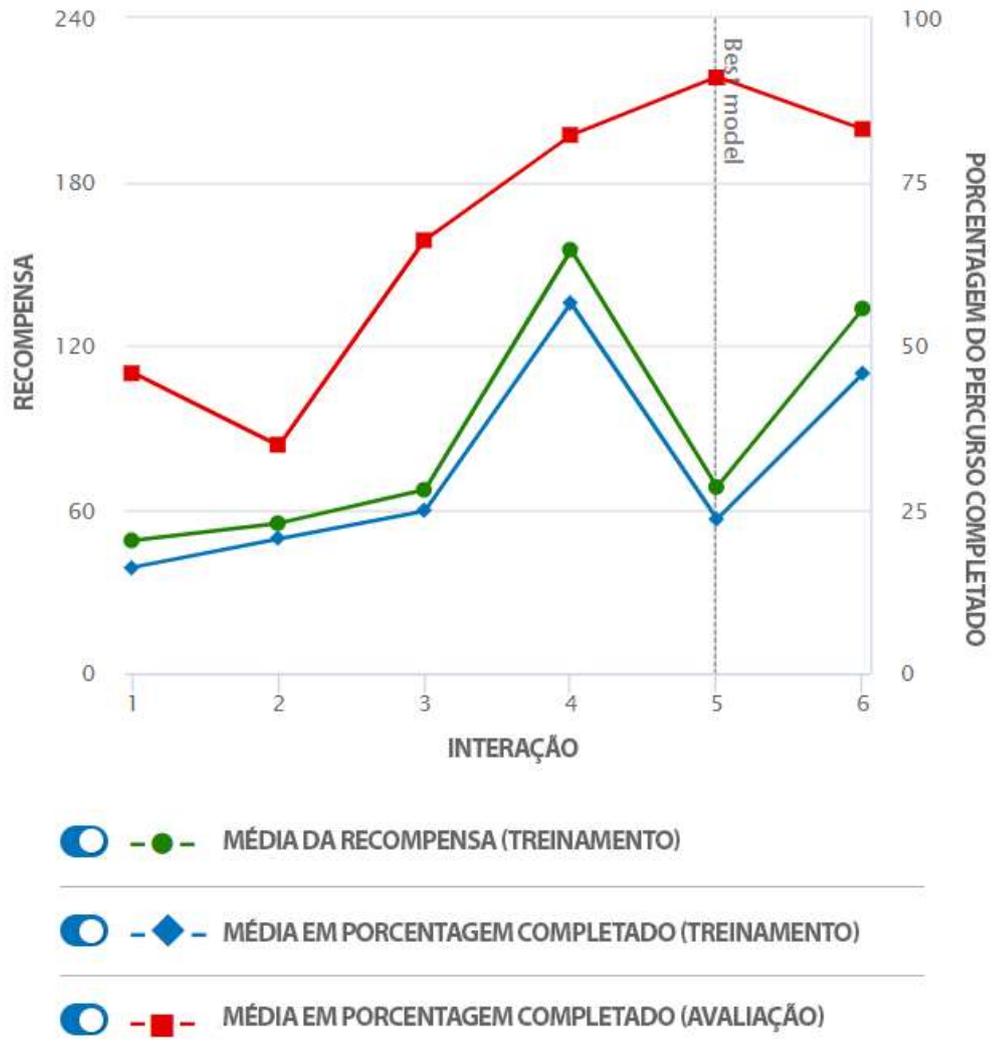
Tentativas	Tempo (MM:SS.mmm)	Resultado das Tentativas (% Percurso completado)	Status
1	00:20.498	74%	Fora do percurso
2	00:06.264	24%	Fora do percurso
3	00:19.877	74%	Fora do percurso

Fonte: Autor

8.3 VERSÃO 3 DO MODELO RL

Na terceira versão, conseguimos resultados satisfatórios em relação às versões anteriores, que podem ser constatadas no gráfico 8.3.1 e na tabela 8.3.1.

Gráfico 8.3.1 - Gráfico da versão 3 do modelo RL



Fonte: Autor

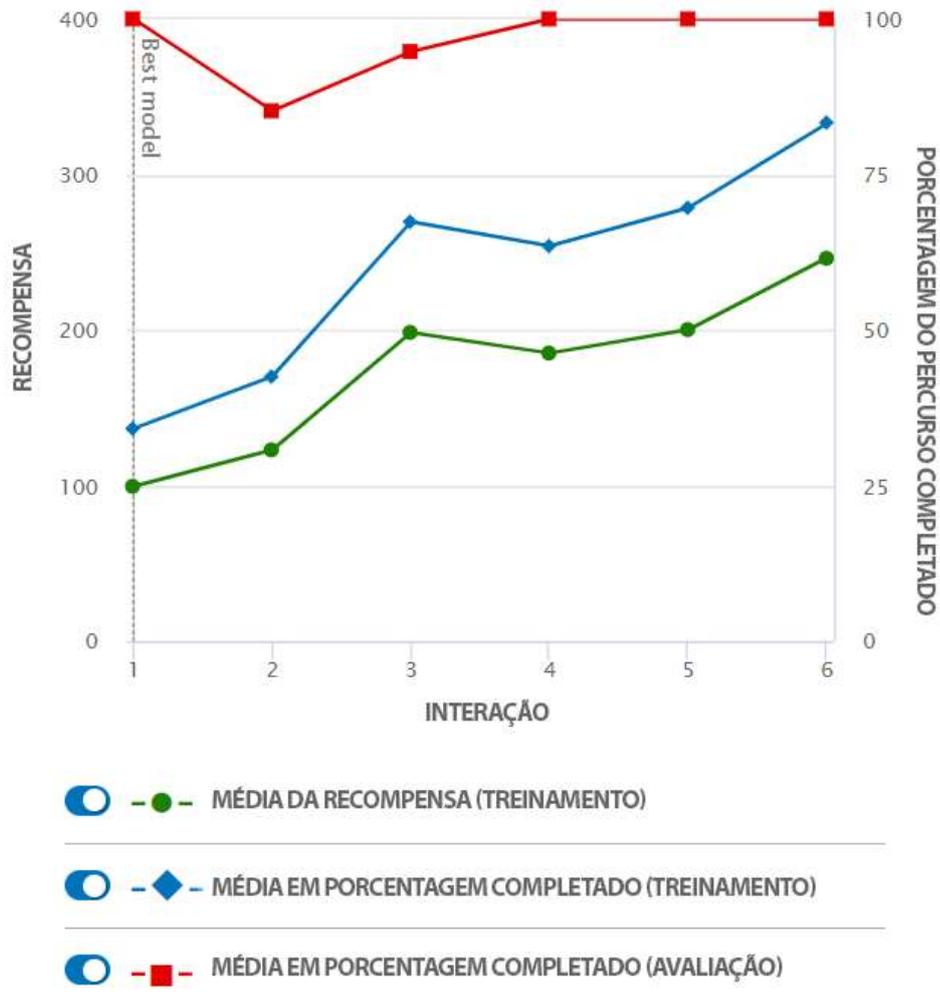
Tabela 8.3.1 - Evolução da versão 3 do modelo RL

Tentativas	Tempo (MM:SS.mmm)	Resultado das Tentativas (% percurso completado)	Status
1	00:24.867	100%	Volta completa
2	00:25.640	100%	Volta completa
3	00:07.484	26%	Fora do percurso
4	00:25.188	100%	Volta completa
5	00:24.772	100%	Volta completa

Fonte: Autor

8.4 VERSÃO 4 DO MODELO RL

Na versão 4 do modelo, constatou-se um aperfeiçoamento no resultado em relação às versões anteriores. Na evolução notou-se que o modelo RL V4 concluiu todas as voltas programadas e, que as dificuldades apresentadas nas versões anteriores, foram sanadas. Pontos específicos que dificultaram a tomada de decisões foram suavizados, não interferindo, assim, no desempenho final da versão 4 do modelo. A tabela a seguir expõe a comparação de desempenho entre as versões.

Gráfico 8.4.1 - Gráfico da versão 4 do modelo RL

Fonte: Autor

Tabela 8.4.1 - Evolução da versão 4 do modelo RL

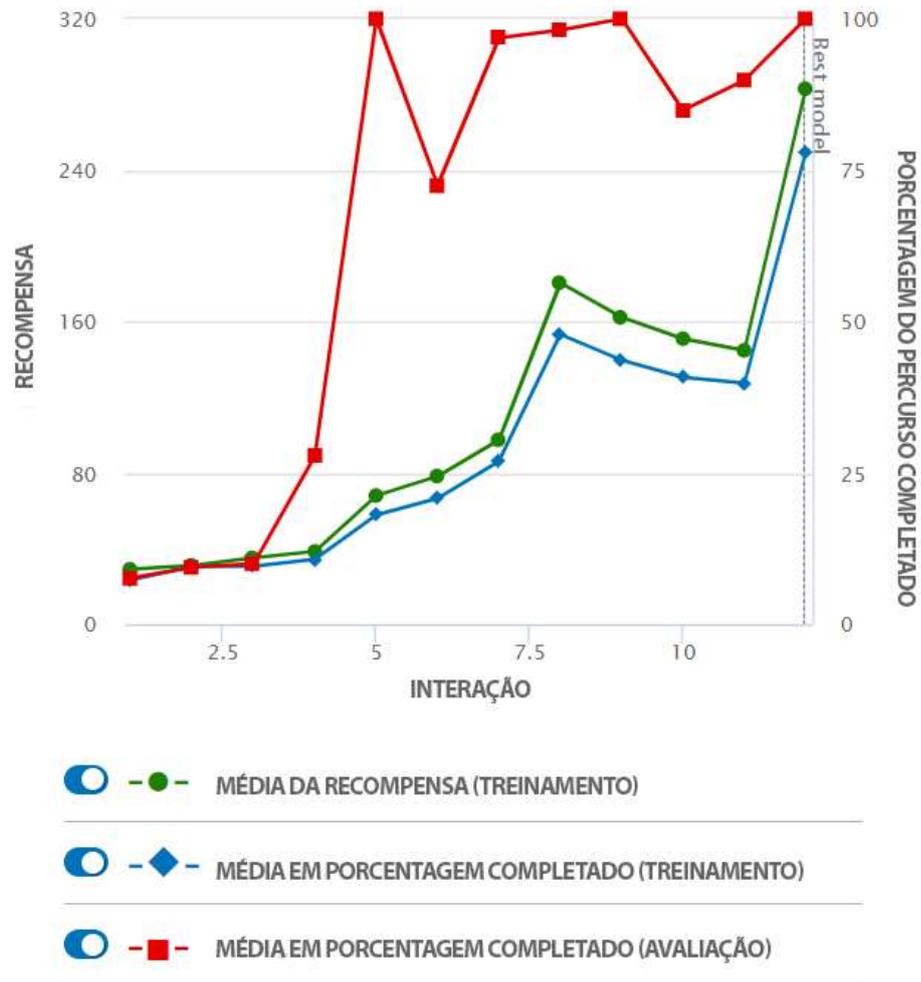
Tentativas	Tempo (MM:SS.mmm)	Resultado das Tentativas (% percurso completado)	Status
1	00:24.319	100%	Volta completa
2	00:24.500	100%	Volta completa
3	00:25.756	100%	Volta completa
4	00:25.070	100%	Volta completa
5	00:24.659	100%	Volta completa

Fonte: Autor

8.5 VERSÃO 5 DO MODELO RL

A quinta versão do modelo alcançou um melhor desempenho em relação às alternativas anteriores, mesmo o teste tendo sido executado em uma pista com maior grau de dificuldade (Fig. 8.5.1). Obviamente, o acréscimo temporal é facilmente compreendido visto a nova pista abarcar uma distância total de 23,12m. A pista inicial contava com apenas 17,6m de distância total em seu percurso. Outro fator preponderante está relacionado ao grau de dificuldade, sendo a última mais complexa em relação à primeira pista.

Podemos notar que o algoritmo consegue desempenhar seu objetivo com uma parcela de facilidade em um determinado tempo (Tab. 8.5.1). Com esta informação, podemos implementar tal algoritmo como forma de salvaguarda quando o SDC estiver com seus sensores não operantes ou não puderem alimentar o feed do SDC a cada interação do seu estado.

Gráfico 8.5.1 - Gráfico da versão 5 do modelo RL

Fonte: Autor

Tabela 8.5.1 - Evolução da versão 5 do modelo RL

Tentativas	Tempo (MM:SS.mmm)	Resultado das Tentativas (% percurso completado)	Status
1	00:29.231	100%	Volta completa
2	00:29.386	100%	Volta completa
3	00:30.586	100%	Volta completa
4	00:29.651	100%	Volta completa
5	00:28.751	100%	Volta completa

Fonte: Autor

Figura 8.5.1 - Track com um grau maior de dificuldade para ser usado no modelo RL V5



Fonte: AWS

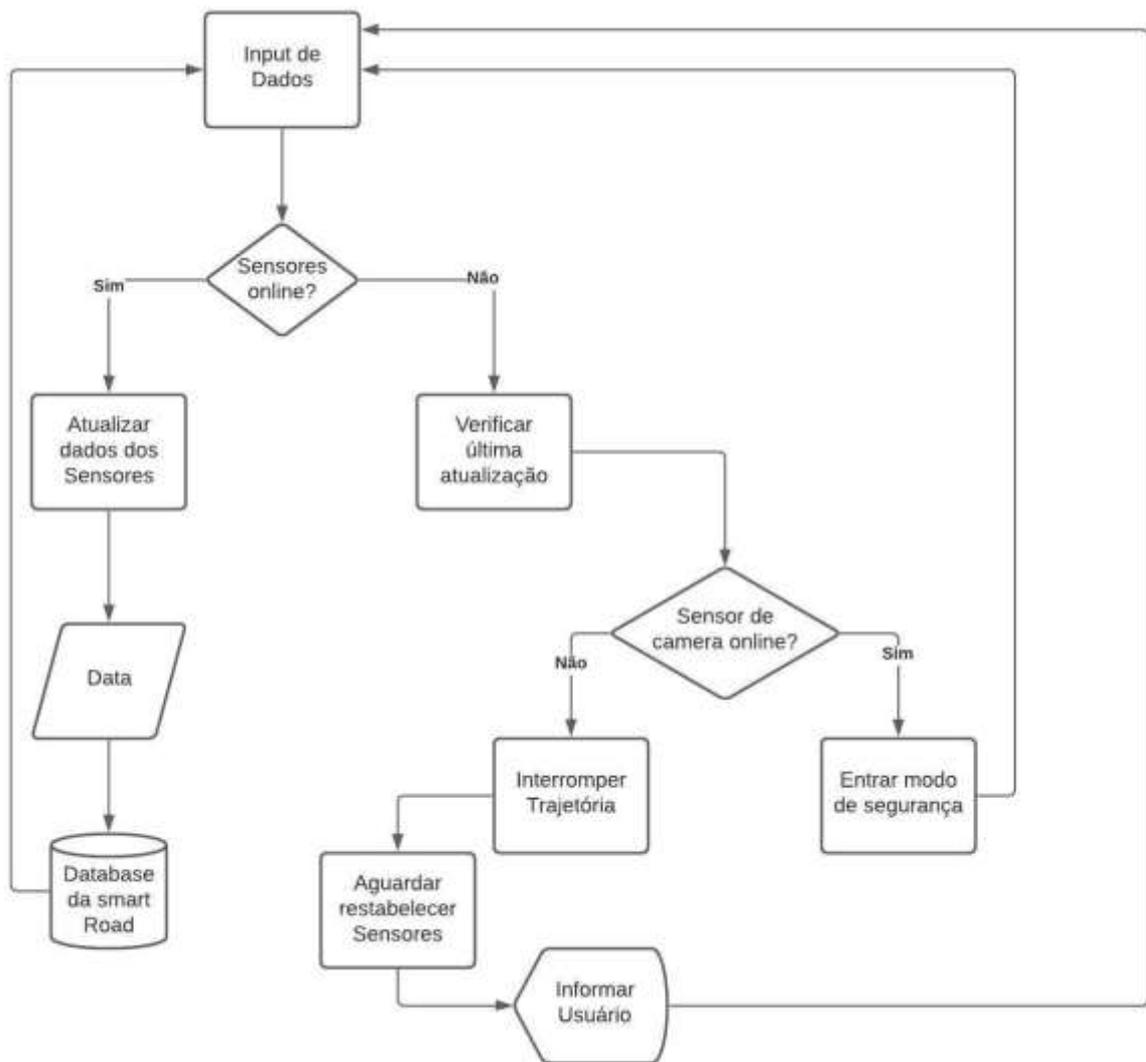
Tabela 8.5.2 - Resultados comparativos das versões dos modelos RL

Versão do Modelo	Tempo (MM:SS.mmm)	(% Trajetória Concluída)	Status
Modelo RL V1	00:29.486	100%	Volta Completa
Modelo RL V2	00:19.877	74%	Volta Incompleta
Modelo RL V3	00:24.772	100%	Volta Completa
Modelo RL V4	00:24.319	100%	Volta Completa
Modelo RL V5	00:28.751	100%	Volta Completa

Fonte: Autor

Dessa forma é possível implementar um algoritmo (Fig.8.5.2 e Apêndice A) de loop, por meio do qual o sistema do carro (Fig. 8.5.3) pode verificar seus sensores e continuar no trajeto, caso contrário, o sistema é forçado a entrar em modo de “segurança”.

Figura 8.5.2 - Fluxograma lógico do acesso ao algoritmo no modo de segurança do modelo RL



Fonte: Autor

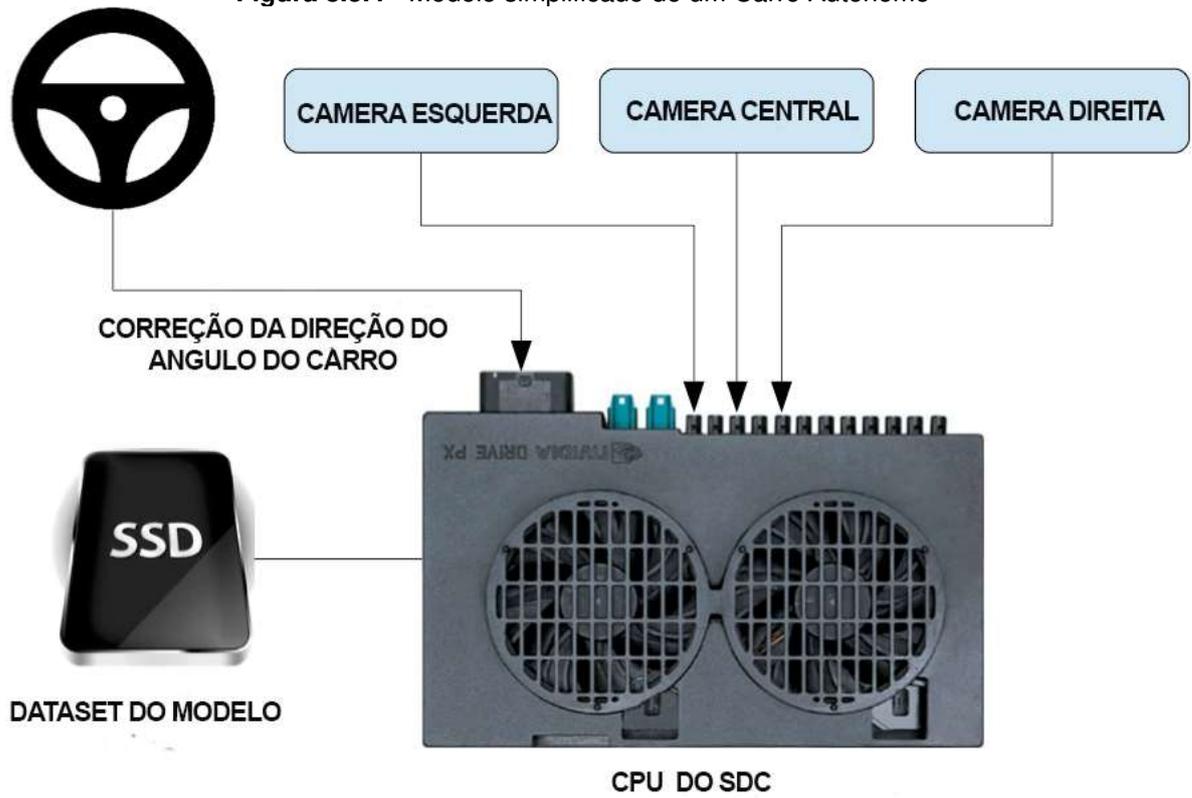
Figura 8.5.3 - ScreenShot da simulação do modelo RL V5 no segundo track no simulador da AWS



Fonte: Autor

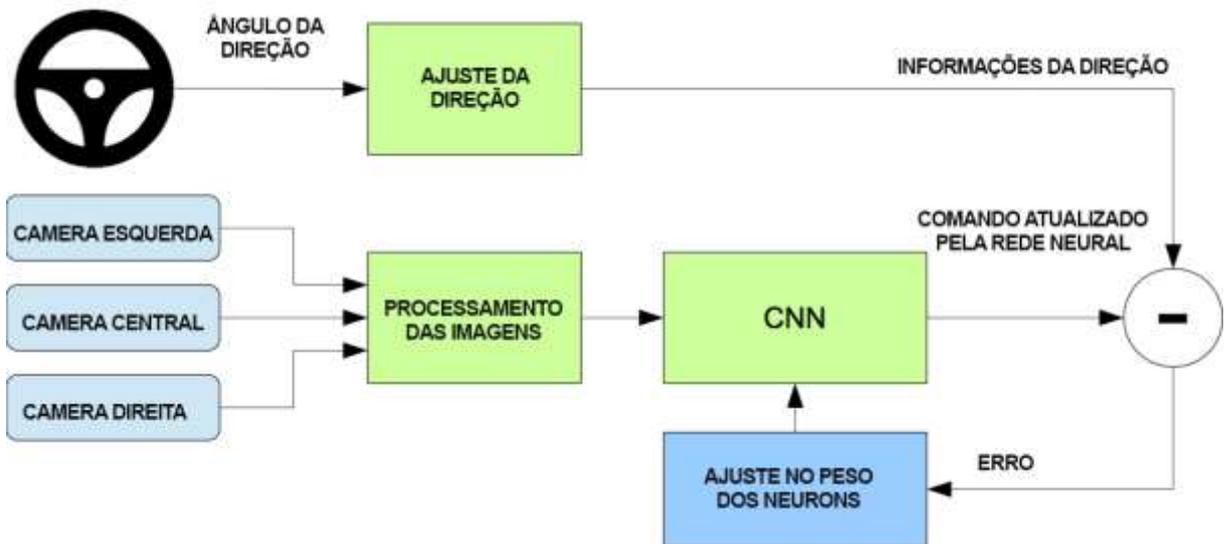
Podemos visualizar a estrutura completa do fluxo de funcionamento de um carro autônomo como mostra a figura 8.5.4 e com mais detalhes na figura 8.5.5

Figura 8.5.4 - Modelo simplificado de um Carro Autônomo



Fonte: Autor

Figura 8.5.5 - Fluxo do modelo do Carro Autônomo



Fonte: Autor

9 CARROS AUTÔNOMOS E O CENÁRIO BRASILEIRO

Depois dos tópicos que foi apresentado nos capítulos anteriores como *smart city* e *smart roads*, a pergunta sobre como será a realidade dos carros autônomos no Brasil continua mais forte do que nunca. Existe muitas barreiras que impedem o avanço do desenvolvimento dos carros autônomos no Brasil.

9.1 ÍNDICE DE PRONTIDÃO PARA VEÍCULOS AUTÔNOMOS 2020

Segundo um estudo feito pela consultoria KPMG (Índice de Prontidão para Veículos Autônomos 2020, 2021) o Brasil segue como o país menos preparado para a tecnologia de mobilidade autônoma. Foi avaliado mais de 30 países, sobre aspectos como infraestrutura, legislação e pesquisas na área. Nessa lista estão países que se destacam na pesquisa sobre carros autônomos como Estados Unidos e Alemanha.

Assim como acontece em muitos casos sobre pesquisas científicas, as leis brasileiras não favoráveis as mudanças que estão acontecendo já em vários países de primeiro mundo. É evidente essa diferença conforme o Estudo da KPMG na figura 9.1.1. Outro fator muito relevante seria o incentivo a pesquisa que o país tem sobre esse assunto, mais uma vez, a classificação do Brasil não é muito melhor como é visto na figura 9.1.2. Sobre o quesito infraestrutura, o país não tem nenhuma perspectiva para se preparar para as mudanças com carros autônomos, como se pode ver na figura 9.1.3

Apesar de todos os contratempos e burocracias que existe no país, o Brasil já tem carros autônomos, o CaRina – carro robótico de navegação autônoma que foi desenvolvido pela equipe do Laboratório de Robótica Móvel da USP, na cidade de São Paulo e testado nas Ruas de São Carlos (Laboratório de Robótica Móvel, 2013).

Um outro projeto brasileiro que se destaca é o IARA - *Intelligent Autonomous Robotic Automobile*, esse projeto foi desenvolvido pelos pesquisadores do Laboratório de Computação de Alto Desempenho (Lcad) da UFES.

Figura 9.1.1 - Pesquisa da KPMG sobre política e legislação em relação a mobilidade autônoma

	Posição	Regulamentação de veículos autônomos	Flotas de veículos autônomos financiadas pelo governo	Agência focada em veículos autônomos	Orientação futura do governo	Eficiência do sistema jurídico em regulamentos complexos	Prontidão do governo para a mudança	Acabamento do acompanhamento de dados	Posição de Pilar 1 (não ajustado)
1	Singapura	1,000	1,000	1,000	1,000	0,673	1,000	0,411	6,064
2	Reino Unido	0,929	0,857	0,857	0,634	0,608	0,780	1,000	5,626
3	Holanda	1,000	0,929	0,714	0,839	0,826	0,780	0,688	5,578
4	Finlândia	1,000	0,857	0,714	0,718	1,000	0,780	0,451	5,521
5	Nova Zelândia	0,929	0,714	0,929	0,573	0,792	0,829	0,743	5,505
6	Estados Unidos	0,857	0,929	0,714	0,763	0,792	0,634	0,771	5,461
7	Alemanha	0,786	0,857	0,857	0,604	0,717	0,829	0,621	5,301
8	Emirados Árabes Unidos	0,857	0,714	0,929	0,880	0,905	0,951	0,081	5,278
9	Canadá	0,786	1,000	0,714	0,502	0,814	0,756	0,870	5,242
10	Noruega	0,929	0,857	0,643	0,575	0,629	0,854	0,674	5,161
11	Áustria	0,857	0,857	0,929	0,502	0,608	0,810	0,620	4,952
12	Dinamarca	0,714	0,643	0,857	0,589	0,666	0,829	0,635	4,931
13	Taiwan	0,857	1,000	0,786	0,334	0,425	0,650	0,060	4,920
14	Frça	0,786	0,857	0,714	0,481	0,615	0,585	0,815	4,854
15	Suécia	0,714	0,714	0,714	0,564	0,624	0,878	0,625	4,834
16	Coreia do Sul	0,857	1,000	0,857	0,488	0,346	0,463	0,796	4,777
17	Austrália	1,000	0,571	0,714	0,409	0,516	0,707	0,765	4,683
18	Japão	0,571	0,857	0,571	0,509	0,642	0,659	0,691	4,496
19	Israel	0,714	0,786	0,643	0,532	0,603	0,488	0,331	4,097
20	Bélgica	0,929	0,714	0,714	0,371	0,585	0,512	0,379	4,024
21	China	0,786	0,929	0,643	0,490	0,435	0,561	0,000	3,944
22	República Tcheca	0,857	1,000	0,714	0,186	0,222	0,512	0,309	3,900
23	Espanha	0,857	0,571	0,714	0,163	0,322	0,317	0,688	3,614
24	Chile	0,429	0,571	0,429	0,435	0,435	0,439	0,346	3,083
25	Hungria	0,643	0,857	1,000	0,268	0,000	0,244	0,046	3,056
26	Rússia	0,571	0,286	0,857	0,367	0,349	0,293	0,390	2,873
27	Itália	0,857	0,643	0,643	0,090	0,096	0,292	0,452	2,843
28	Índia	0,000	0,000	0,000	0,336	0,514	0,341	0,288	1,679
29	México	0,143	0,143	0,143	0,168	0,194	0,098	0,670	1,557
30	Brasil	0,286	0,143	0,143	0,071	0,119	0,000	0,488	1,190

Fonte: Índice de Prontidão para Veículos Autônomos 2020, pág.53

Figura 9.1.2 - Pesquisa da KPMG sobre tecnologia e inovação em relação a mobilidade autônoma

Posição	País	Presença de indústrias	Sedes das empresas de tecnologia de veículos autônomos	Patentes relacionadas a veículos autônomos	Investimentos da indústria em veículos autônomos	Disponibilidade de tecnologias mais modernas	Capacidade de inovação	Cibersegurança	Avaliação da computação em nuvem, IA e IoT	Participação do mercado de veículos elétricos	Pontuação do Pilar 2 (não ajustada)
1	Israel	0,750	1,000	0,052	1,000	0,946	0,716	0,679	0,351	0,029	5,722
2	Estados Unidos	1,000	0,122	0,296	0,370	0,931	0,939	0,989	1,000	0,033	5,681
3	Japão	0,917	0,022	1,000	0,055	0,843	0,808	0,889	0,707	0,017	5,258
4	Alemanha	1,000	0,078	0,849	0,124	0,751	1,000	0,822	0,574	0,052	5,250
5	Noruega	0,917	0,053	0,012	0,000	0,971	0,576	0,915	0,764	1,000	5,209
6	Suécia	0,833	0,203	0,352	0,061	0,937	0,826	0,737	0,805	0,201	4,946
7	Coreia do Sul	1,000	0,026	0,856	0,023	0,633	0,826	0,874	0,351	0,043	4,832
8	Finlândia	0,833	0,171	0,017	0,026	1,000	0,752	0,837	0,708	0,123	4,475
9	Reino Unido	0,833	0,104	0,113	0,011	0,855	0,806	1,000	0,676	0,057	4,458
10	Holanda	0,667	0,066	0,032	0,103	0,907	0,763	0,900	0,701	0,265	4,403
11	Singapura	0,833	0,132	0,020	0,004	0,771	0,738	0,928	0,717	0,085	4,230
12	Frância	0,833	0,042	0,116	0,029	0,735	0,783	0,972	0,567	0,049	4,127
13	Canadá	1,000	0,085	0,012	0,073	0,782	0,711	0,915	0,458	0,047	4,114
14	Taiwan	0,833	0,007	0,094	0,000	0,561	0,851	0,856	0,736	0,018	3,946
15	Dinamarca	0,667	0,015	0,011	0,000	0,740	0,761	0,829	0,800	0,034	3,896
16	Áustria	0,667	0,017	0,036	0,044	0,685	0,722	0,772	0,450	0,065	3,527
17	Austrália	0,500	0,034	0,048	0,007	0,576	0,609	0,911	0,545	0,020	3,248
18	Bélgica	0,417	0,032	0,007	0,001	0,808	0,652	0,746	0,521	0,057	3,242
19	Nova Zelândia	0,667	0,019	0,010	0,000	0,743	0,409	0,892	0,567	0,048	3,155
20	China	1,000	0,002	0,045	0,014	0,023	0,503	0,777	0,446	0,103	2,913
21	Itália	0,833	0,008	0,012	0,000	0,200	0,519	0,796	0,200	0,016	2,875
22	Emirados Árabes Unidos	0,833	0,008	0,000	0,005	0,787	0,221	0,731	0,193	0,085	2,864
23	Espanha	0,600	0,016	0,013	0,000	0,462	0,400	0,924	0,338	0,025	2,769
24	Hungria	0,667	0,037	0,006	0,026	0,371	0,111	0,742	0,193	0,033	2,095
25	República Tcheca	0,667	0,007	0,008	0,000	0,543	0,325	0,215	0,326	0,000	2,025
26	Rússia	0,333	0,004	0,007	0,001	0,000	0,235	0,794	0,056	0,001	1,432
27	Chile	0,333	0,004	0,001	0,000	0,554	0,000	0,000	0,180	0,001	1,084
28	Índia	0,167	0,001	0,001	0,000	0,122	0,190	0,540	0,000	0,000	1,020
29	México	0,000	0,000	0,001	0,000	0,209	0,025	0,345	0,122	0,001	0,763
30	Brasil	0,167	0,001	0,001	0,000	0,046	0,144	0,232	0,144	0,001	0,736

Fonte: Índice de Prontidão para Veículos Autônomos 2020, pág.57

Figura 9.1.3 - Pesquisa da KPMG sobre infraestrutura em relação a mobilidade autônoma

Posição	País	Estações de carregamento de veículos autônomos	Cobertura 4G	Qualidade das estradas	Prontidão para a mudança na infraestrutura tecnológica	Velocidade de conexão móvel (metade do peso)	Banda larga (metade do peso)	Pontuação do Pilar 3 (não ajustada)
1	Holanda	1,000	0,832	0,993	0,823	0,765	0,792	4,221
2	Coreia do Sul	0,060	1,000	0,838	0,688	0,959	0,917	3,525
3	Noruega	0,008	0,920	0,448	0,467	0,728	0,958	3,495
4	Emirados Árabes Unidos	0,010	0,636	0,869	1,000	1,000	0,823	3,431
5	Singapura	0,095	0,729	0,729	0,795	0,578	1,000	3,379
6	Japão	0,078	0,957	0,894	0,689	0,772	0,958	3,233
7	Áustria	0,166	0,611	0,871	0,844	0,438	0,708	3,095
8	Suécia	0,290	0,771	0,608	0,578	0,473	0,958	3,023
9	Estados Unidos	0,070	0,839	0,714	0,600	0,393	0,917	2,878
10	Dinamarca	0,158	0,662	0,744	0,556	0,491	0,875	2,823
11	Finlândia	0,098	0,714	0,657	0,641	0,493	0,823	2,738
12	Austrália	0,010	0,743	0,557	0,578	0,693	1,000	2,735
13	Canadá	0,014	0,689	0,587	0,276	0,788	0,917	2,560
14	Taiwan	0,024	0,588	0,754	0,533	0,483	0,860	2,558
15	Espanha	0,062	0,639	0,782	0,533	0,327	0,750	2,505
16	Reino Unido	0,141	0,543	0,338	0,689	0,313	0,750	2,442
17	Frância	0,150	0,364	0,704	0,533	0,467	0,792	2,381
18	Bélgica	0,182	0,746	0,399	0,333	0,516	0,708	2,283
19	Alemanha	0,165	0,264	0,606	0,600	0,329	0,667	2,182
20	Nova Zelândia	0,021	0,250	0,420	0,711	0,522	0,917	2,121
21	Hungria	0,023	0,782	0,203	0,333	0,433	0,542	1,919
22	China	0,079	0,981	0,456	0,267	0,751	0,250	1,884
23	Itália	0,024	0,339	0,406	0,622	0,318	0,629	1,863
24	República Tcheca	0,033	0,754	0,261	0,289	0,496	0,542	1,856
25	Israel	0,108	0,000	0,337	0,578	0,146	0,833	1,712
26	Chile	0,002	0,257	0,638	0,422	0,117	0,542	1,648
27	Rússia	0,001	0,157	0,138	0,622	0,117	0,629	1,297
28	México	0,007	0,366	0,434	0,200	0,218	0,333	1,284
29	Índia	0,000	0,764	0,437	0,000	0,000	0,000	1,202
30	Brasil	0,001	0,089	0,000	0,311	0,171	0,417	0,696

Fonte: Índice de Prontidão para Veículos Autônomos 2020, pág.61

10 CONCLUSÕES

É inegável o fato de que carros autônomos transformarão o modelo de mobilidade humana apresentado atualmente. Além disso, vamos ter um grande impacto na economia e na logística mundial. A adoção em escala global dos SDCs conduzirá a uma mudança sem precedentes no âmbito econômico, social e ambiental.

Para a população, a independência ao viajar será viável a jovens, idosos, pessoas com impedimentos físicos ou visuais. Estima-se que a redução de congestionamento nas rodovias irá trazer novos estímulos para a economia. Os acidentes automobilísticos fatais serão evitados, este aspecto será, sem dúvidas, o mais notório.

Atualmente a obtenção e manutenção de carros é bastante onerosa. No futuro, carros autônomos circularão em *smart roads* integradas completamente às *smart cities*. Provavelmente a comodidade, causada pelo baixo custo e a alta qualidade apresentados por esse meio de transporte, afete a forma como a sociedade vai encarar esse novo modo de locomoção.

10.1 UM PASSO PARA O FUTURO

Empresas de transportes, como a Uber, possuirão uma frota gigantesca de carros autônomos tornando o serviço acessível. Os veículos movidos a energia elétrica, que nas *smart cities* é um recurso renovável e de baixo custo, tornará o serviço popular. Outro ponto positivo será a facilidade de solicitar um carro ao toque no aplicativo do celular. Provavelmente não será compensatório possuir carro próprio.

Outro importante fator a ser considerado refere-se à redução drástica de emissões de poluentes na atmosfera. Os carros SDCs são produzidos com o intuito de reduzir as emissões de gases ao máximo, esse elemento os converte em ótima iniciativa na redução da contaminação atmosférica, grande fator de preocupação, especialmente em grandes centros urbanos.

As cidades hoje são ambientes desenvolvidos para comportar o uso de carros, dessa maneira resultam cada dia mais hostis para nós, pedestres. Com o crescimento

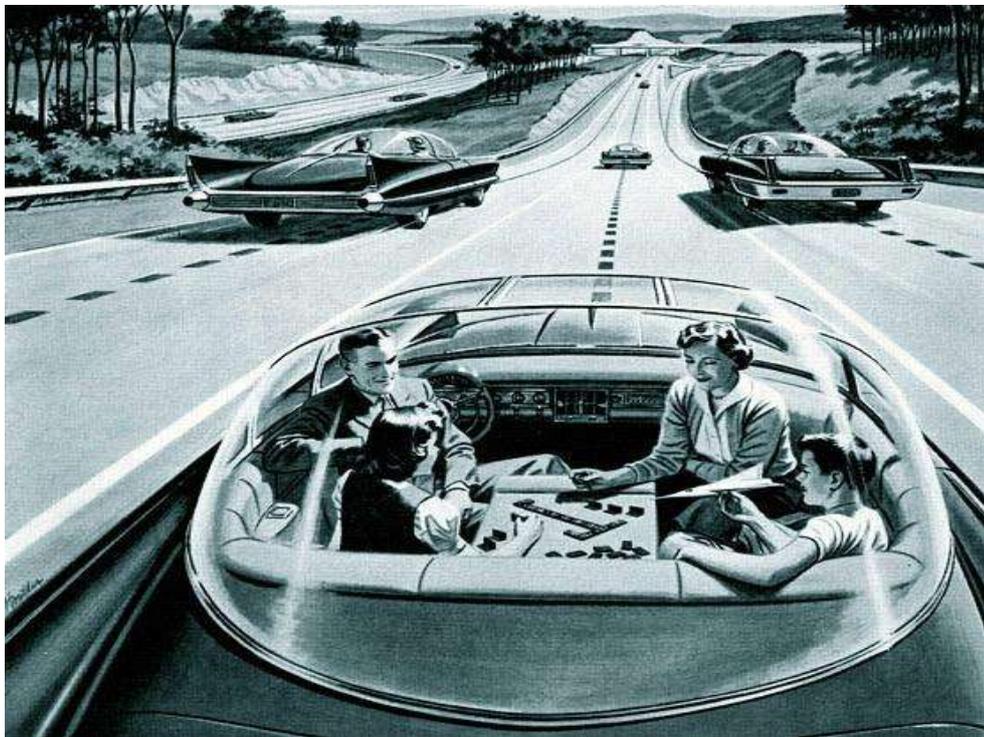
do conceito das *smart cities*, *smart roads* e SDC é possível imaginar uma alteração deste paradigma em relação às urbes.

Devido aos carros autônomos serem precisos em suas trajetórias, não haverá necessidade de estradas largas. Por sua vez, com ruas e estradas mais estreitas, o espaço nas metrópoles será ampliado para usufruto daqueles que mais importam, os moradores. Proporcionando, assim, maior integração entre cidade, pedestre e meio de transporte inteligente.

O ato de atravessar ruas se tornará algo fácil, pois não existirá o fator emocional nos carros autônomos. Ao invés de prestar atenção na direção no percurso, o tempo usado para a locomoção de um ponto a outro pode ser aproveitado de outras formas. Fatores como esses transformarão a percepção desse novo modelo de transporte.

Essa idealização não é nova, no começo de 1950, apareceu com descrições detalhadas na revista *Popular Mechanics* (Fig. 10.1.1) de como seria o futuro com carros autônomos (POCZTER e JANKOVIC, 2014).

Figura 10.1.1 - Propaganda em 1956 de como seriam os carros autônomos do futuro



Fonte: www.engineering.com/story/the-road-to-driverless-cars-1925---2025

Muitas empresas estão se preparando para mudar a logística de seus produtos (TARUTIN, 2018). Uma das pioneiras é a *Google*, que começou esse empreendimento oficialmente com a empresa Waymo em 2009, testando vários modelos de carros em pequenas cidades do interior dos USA e UK. No momento, conta com serviços avançados oferecidos em algumas cidades, como carros para transporte de passageiros e caminhões autônomos para entrega de produtos, vistos nas figuras 10.1.2 e 10.1.3 (ZHAO, LIANG e CHEN, 2017).

Outra grande investidora de recursos neste setor é a Uber (Fig. 10.1.4), que fez parceria com o Centro de Engenharia de Robótica Nacional na Universidade Carnegie Mellon. Apesar do incidente envolvendo a gigante que ocasionou uma vítima fatal, são nítidos seus objetivos para o futuro.

Figura 10.1.2 - Projeto do Google de Carros Autônomos que se iniciou em 2009



Fonte: <https://waymo.com>

Figura 10.1.3 - Projeto do Google Waymo Via na área de logística



Fonte: <https://waymo.com/waymo-via>

Figura 10.1.4 - Projeto piloto da Empresa Uber com carros autônomos



Fonte: <https://startupi.com.br/2017/09/camara-dos-eua-aprova-projeto-sobre-carros-autonomos/uber-self-driving-car/>

A startup Nuro, fundada por engenheiros que trabalharam no projeto de carro autônomo da Google, está desenvolvendo pequenos veículos autônomos específicos para entregas. A franquia de pizzaria Domino's (Fig. 10.1.5 e Fig. 10.1.6) em parceria com a Nuro começou a fazer entregas em certas áreas da cidade de Houston desde 2016.

Figura 10.1.5 - Projeto da Pizzaria Domino's de entrega usando carros autônomos em parceria com a startup Nuro



Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

Figura 10.1.6 - Veículo autônomo da startup Nuro sendo carregado para entrega de pizzas da Domino's



Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

Apesar desse vislumbre em relação aos carros autônomos, podemos concluir que este só será possível quando houver total integração entre *smart roads* e *smart cities*, sem falhas neste círculo fechado. O maior problema se encontrará no período de transição, no qual carros autônomos circularão com carros tradicionais, o que pode gerar desconforto nas pessoas, visto que muitos se questionam acerca da segurança dos mesmos. Tal incredulidade concentra-se em países sem recursos suficientes para fazer uma transição rápida e confiável. Para isso é necessário considerar fortemente modelos de *Deep Learning* que consigam se adaptar às condições adversas (Fig. 10.1.7).

Figura 10.1.7 - Visão de como vai ser o transporte em Londres em 2045



Fonte: <https://www.businessinsider.com/6-predictions-about-the-future-of-transportation-2015-11>

11 TRABALHOS FUTUROS

Com a conclusão desse trabalho, pode-se perceber um campo vasto a ser estudado da Inteligência Artificial.

Em função da falta de acesso a recursos e um tempo muito limitado para um assunto muito complexo, recomenda-se para trabalhos futuros as seguintes propostas de estudos:

- Estudo de modelos matemáticos específicos de *Deep Learning* para carros autônomos aplicados à visão computacional.

- Estudo de modelos de Aprendizagem de Reforço integrado com sensores de reconstrução espacial em 3D como o sensor LIDAR.

- Estudo de modelos de *Deep Learning* de comunicação do carro autônomo com o seu ambiente.

- Desenvolvimento de modelos de Inteligência Artificial para carros autônomos aplicados para entregas de produtos.

Por fim, sugere-se também a incorporação da incerteza nos parâmetros do modelo desenvolvido neste trabalho, como por exemplo o parâmetro de correção de direção. Essas incertezas poderiam ser melhor desenvolvidas em uma programação mais robusta em ambiente físico.

12 REFERÊNCIAS

- "PHANTOM Auto" to Be Operated Here. **The Free-Lance Star**, 17 Junho 1932. 1.
- 2020 Global Automotive Consumer Study Is consumer interest in advanced automotive technologies on the move? Deloitte Development LLC. London, p. 28. 2020.
- AIROLA, R.; HAGER, K. **Image Classification, Deep Learning and Convolutional Neural Networks**. Karlstad: Karlstad University. 2017. p. 87.
- AMATO, J. **DARPA DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 1958–2018**. Tampa: Faircount Media Group, 2018.
- ANDERSON, J. et al. **Autonomous Vehicle Technology A Guide for Policymakers**. Santa Monica: RAND Corporation, 2016.
- ANTONELLO, J. Introdução a Visão Computacional com Python e OpenCV. **Instituto Federal Caratinense**, 2017. 2-68.
- BALLARD, D.; BROWN, C. **Computer Vision**. 1ª. ed. New Jersey: Prentice-Hall, 1982.
- BLOOM, et al. Self-Driving Cars and Data Collection: Privacy Perceptions of Networked Autonomous Vehicles. **Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)**, 12 Julho 2017. 357-375.
- BOJARSK, M. et al. End to End Learning for Self-Driving Cars. **Computer Vision and Pattern Recognition**, Cornell, 25 Abril 2016. 9.
- CAPGEMINI RESEARCH INSTITUTE. **The Autonomous Car: A Consumer Perspective**. [S.l.]. 2018.
- CHEN, et al. A Vision of C-V2X: Technologies, Field Testing and Challenges with Chinese Development. **arXiv**, 2002.
- CHERUVU, R. Big Data Applications in Self-Driving Cars. **Research Gate**, Dezembro 2015. Disponível em: <https://www.researchgate.net/publication/318940413_Big_Data_Applications_in_Self-Driving_Cars>.
- CLEAN Energy Production through Roadway Surface Power Harvesting Systems for Increased Safety in the Transportation Sector. **cordis**, 2013. Disponível em: <<https://cordis.europa.eu/article/id/92180-onroad-energy-harvesting>>. Acesso em: 15 fevereiro 2021.
- DELCKER, J. The man who invented the self-driving car (in 1986) :Long before Big Tech got behind the wheel, Ernst Dickmanns unleashed a driverless Mercedes onto the roads of Europe. **Politico**, 19 Julho 2018. Disponível em: <<https://www.politico.eu/article/delf-driving-car-born-1986-ernst-dickmanns-mercedes/>>.
- DEY, C. et al. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network – Performance evaluation. **Transportation Research Part C: Emerging Technologies**, v. 68, p. 168-184, 2016.
- DUDA, R. O.; HART, P. E. **Use of the Hough Transformation to Detect Lines and Curves in Pictures**. 1ª. ed. [S.l.]: ACM, v. 15, 1972.

FUKUSHIMA, K. Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition. **Neural Network**, v. 1, p. 119-130, 1988.

GELDIEV, E. M.; NENKOV, N. V.; PETROVA, M. M. EXERCISE OF MACHINE LEARNING USING SOME PYTHON TOOLS AND TECHNIQUES. **CBU INTERNATIONAL CONFERENCE ON INNOVATIONS IN SCIENCE AND EDUCATION**, 21 Março 2018. 1062 - 1070.

GIBBS, T. Reaching New Cyber 5 Revenue Records: Inspiring E-Commerce Success Stories. **Channel Advisor**, 2020. Disponível em: <<https://www.channeladvisor.com/>>. Acesso em: 11 Setembro 2020.

GOMES, D. S. Inteligência Artificial: Conceitos e Aplicações. <http://www.olharcientifico.kinghost.net>, 2010. 234-246.

GONZAGA, L. A. **APLICAÇÕES DA VISÃO COMPUTACIONAL UTILIZANDO PYTHON**. APPLICATIONS OF COMPUTER VISION USING PYTHON. Uberlândia: Universidade Federal de Uberlândia. 2017. p. 118.

GOUVÊA, B. V. AS ESTRATÉGIAS DO GOOGLE E DA GENERAL MOTORS PARA O CARRO AUTÔNOMO. **pantheon**, Setembro 2016. Disponível em: <<https://pantheon.ufrj.br/bitstream/11422/5029/1/Breno%20Vieira%20de%20Gouv%20de%20AAa%20-%20As%20Estrat%20gias%20do%20Google%20e%20da%20General%20Motors%20para%20o%20Carro%20Aut%20B4nomo.pdf>>.

HILL, D.; AGARWA, A.; TONG, N. **ASSESSMENT OF PIEZOELECTRIC MATERIALS FOR ROADWAY ENERGY HARVESTING**. California Energy Commission. Oakland, p. 106. 2014. (CEC-500-2013-007).

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. **Neural Computation**, v. 18, n. 7, p. 1527 - 1554, 2006.

HOSTER, H. Wired-up roads will soon charge your electric car – while you’re driving. **the conversation**, 8 fevereiro 2017. Disponível em: <<https://theconversation.com/wired-up-roads-will-soon-charge-your-electric-car-while-youre-driving-72625>>. Acesso em: 02 março 2021.

ÍNDICE de Prontidão para Veículos Autônomos 2020. **KPMG**, 2021. Disponível em: <<https://assets.kpmg/content/dam/kpmg/br/pdf/2020/09/autonomous-vehicles-readiness-index-2020.pdf>>. Acesso em: 23 Maio 2021.

KOTTASOVA, I.; PETROFF, J. These roads will charge cars as they drive. **Money.com**, 15 agosto 2015. Disponível em: <<https://money.cnn.com/2015/08/18/technology/uk-electric-cars-roads/index.html>>. Acesso em: 03 fevereiro 2021.

KRÖGER, F. **Autonomous Driving**. Paris: Institut d’histoire moderne et contemporaine (IHMC), 2016.

LABORATÓRIO de Robótica Móvel. **Laboratório de Robótica Móvel**, 2013. Disponível em: <<http://irm.icmc.usp.br/web/index.php?n=Port.ProjCarina2Info>>. Acesso em: 15 Abril 2021.

LARSON, G. A. Knight Rider. **IMDB**. Disponível em: <https://www.imdb.com/title/tt0083437/?ref_=fn_al_tt_1>. Acesso em: 02 julho 2020.

LUTIN, J.; KORNHAUSER, J. The Revolutionary Development of Self-Driving Vehicles and Implications for the Transportation Engineering Profession. **Ite Journal**, Julho 2013. 28-32.

MINGHUANG, C. et al. Vehicle-to-Infrastructure (V2I) offloading from cellular network to 802.11p Wi-Fi network based on the Software-Defined Network (SDN) architecture. **Vehicular Communications**, julho 2017. 288-300.

- MIOVISION TrafficLink hardware. **Miovision**, 2021. Disponível em: <<https://miovision.com/trafficlink/hardware/>>. Acesso em: 03 fevereiro 2021.
- MOAVENZADEH, J.; LANG, N. S. Reshaping Urban Mobility with Autonomous Vehicles Lessons from the City of Boston. **World Economic Forum**, p. 3 - 33, 2018.
- MOUYAL, N. Harvesting energy from roads - New technologies capture ambient energy and convert it to electric power. **etech**, 15 março 2018. Disponível em: <<https://etech.iec.ch/issue/2018-02/harvesting-energy-from-roads>>. Acesso em: 03 março 2021.
- NEU, C. V. Desenvolvimento de um sistema de reconhecimento de sinais de trânsito utilizando. **Unisc**, 2014. 1-79.
- OPENCV. OpenCV Documentation. **OpenCV**. Disponível em: <<https://docs.opencv.org/>>. Acesso em: 20 Agosto 2019.
- ORLANDINI,. Desenvolvimento de Aplicativos Baseados em Técnicas de Visão Computacional para Robô Móvel Autônomo. **Universidade Metodista de Piracicaba**, 2012. 1-156.
- PINDARWATI,. SMART Highway: Near Future Highway Systems in Jakarta. **The Third Asia Future Conference**, 2016.
- POCZTER, S. L.; JANKOVIC, L. M. The Google Car: Driving Toward A Better Future? **Journal of Business Case Studies**, 2014. 8.
- PRAKASH, V. How a driverless car sees the road. **engineers garage**, 2 Janeiro 2021. Disponível em: <<https://www.engineersgarage.com/tech-articles/how-a-driverless-car-sees-the-road/#:~:text=How%20it%20sees%20the%20road,vehicles%2C%20pedestrians%20in%20the%20surrounding.>>. Acesso em: 10 Janeiro 2021.
- RÖSSLER, B. **Requirements for intersection safety applications**. Intersafe. [S.l.], p. 54. 2007.
- SCHULMAN, J. et al. Proximal Policy Optimization Algorithms. **arXiv**, 28 Agosto 2017.
- SMART TRANSPORTATION ALLIANCE. Smart Roads: A Vision. **transportation.org**, 2015.
- SMITH,. Computer Science & Engineering, 2012. Disponível em: <https://www.cse.unr.edu/~bebis/CS485/Lectures/Intro_OpenCV.pdf>. Acesso em: 15 junho 2019.
- SUTTON, S. R.; BARTO, G. A. **Reinforcement Learning, second edition: An Introduction**. [S.l.]: MIT Press, 2018.
- SZEGEDY, C. et al. **Going Deeper with Convolutions**. THE COMPUTER VISION FOUNDATION. Boston: [s.n.]. 2015.
- TARUTIN,. SELF-DRIVEN VEHICLES IN LOGISTICS: TODAY AND THE FUTURE. **SAVONIA UNIVERSITY OF APPLIED SCIENCES**, 24 Janeiro 2018. 37.
- TENSORFLOW , 2015. Disponível em: <<https://www.tensorflow.org/>>. Acesso em: 15 junho 2019.
- TESLA Inc. Disponível em: <<https://www.tesla.com>>. Acesso em: 19 Setembro 2019.
- TOH, C. K. et al. Wireless Digital Traffic Signs of The Future. **The Institution of Engineering and Technology**, Wales, janeiro 2019. 7.

TOH, K. et al. Advances in smart roads for future smart cities. **Proceedings of the Royal Society**, Londres, 22 Janeiro 2020.

TURING, A. M. COMPUTING MACHINERY AND INTELLIGENCE. **Mind**, 1950. 433-460.

UNDERSTANDING the Internet of Things (IoT). Londres: GSMA, 2014.

UNITED STATES DEPARTMENT OF TRANSPORTATION. 2018 Fatal Motor Vehicle Crashes: Overview. **United States Department of Transportation**, Washington, 2019. Disponivel em: <<https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812826>>. Acesso em: 10 dezembro 2020.

VIOLA, P.; JONES, M. Rapid Object Detection using a Boosted Cascade of Simple Features, Cambridge, 2001. 1-9.

WAYMO , 2009. Disponivel em: <<https://waymo.com/>>. Acesso em: 20 Setembro 2019.

WORLD'S first electrified road for charging vehicles opens in Sweden. **The Guardian**, 2018. Disponivel em: <<https://www.theguardian.com/environment/2018/apr/12/worlds-first-electrified-road-for-charging-vehicles-opens-in-sweden>>. Acesso em: 03 Fevereiro 2021.

YANN LeCun Interview. **Coursera**. Disponivel em: <<https://www.coursera.org/lecture/convolutional-neural-networks/yann-lecun-interview-4PnfT>>. Acesso em: 15 fevereiro 2021.

YOUSEF, J. **Image Binarization using Otsu Thresholding Algorithm**. Ontario: University of Guelph. 2011. p. 4.

ZHAO, J.; LIANG, B.; CHEN, Q. The key technology toward the self-driving car. **Emerald Insight**, Shenzhen, 2017. 19.

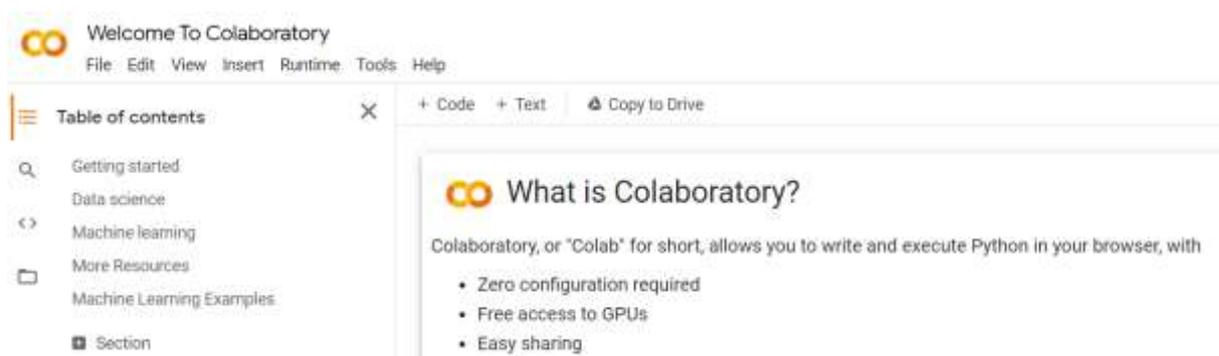
13 ANEXO I – FERRAMENTAS PARA O DESENVOLVIMENTO DO MODELO DL

O Google *Collaboratory* ou mais conhecido como Google Colab é uma ferramenta que a maior empresa de pesquisa no meio digital, a Google, criou para facilitar o desenvolvimento de codificação de *machine learning*. Essencialmente podemos categorizar o Google Colab como um armazenamento na nuvem de notebook ou códigos feito na linguagem python. Todo o serviço do Google Colab pode ser acessado via navegador usando os recursos computacionais dos servidores do Google.

13.1 GOOGLE COLAB

Esse serviço do Google é baseado na mesma tecnologia do jupyter notebook, um aperfeiçoamento do projeto IPHYTON de 2014 feito por Fernando Pérez. O nome desse projeto se deve às três principais linguagens que o sistema suporta sendo Julia, Python e R, e uma homenagem a Galileu Galilei por ter descoberto as luas de Júpiter. Diferente do sistema Júpiter, o Colab pode ser usado exclusivamente online pelo link colab.research.google.com. O único requisito para o uso, é o usuário ter uma conta ativa na plataforma do google (Fig.13.1.1). O acesso ao uso do Colab é muito intuitivo, já que o maior propósito seria o uso acadêmico. Pode-se criar notebooks com códigos e comentários em cada seção para uma melhor interpretação do algoritmo a ser desenvolvido.

Figura 13.1.1 - - Interface inicial do Google Colab



Fonte: Autor

13.2 AWS DEEP RACER

Uma outra ferramenta bastante usada nesse projeto é o *AWS Deep Racer*. Esse é um serviço recente da Amazon Web Services onde a empresa oferece um modelo na escala de 1/18 de um carro autônomo com processador específico (Fig. 13.2.1; 13.2.2; 13.2.3) para o *deploy* de código de modelos DL. Existe também um modelo com sensores mais completos que inclui sensor LIDAR e duas câmeras com visão estéreo (Fig. 13.2.4; 13.2.5). Uma outra facilidade que a empresa Amazon oferece é o simulador virtual, onde o usuário pode desenvolver seus algoritmos e testar virtualmente, (Fig. 13.2.9; 13.2.10; 13.2.11) para consolidar o modelo, antes de fazer o *deploy* no carro físico. Nesse serviço é possível testar algoritmos de RL em diversas pistas.

Figura 13.2.1 - Carro autônomo da AWS visão lateral



Fonte: AWS

Figura 13.2.2 - Visão frontal do carro autônomo da AWS



Fonte: AWS

Figura 13.2.3 - Visão lateral do carro autônomo da AWS



Fonte: Autor

Tabela 13.2.1 - Configuração do carro autônomo da AWS

Car	18th scale 4WD with monster truck chassis
CPU	Intel Atom™ Processor
Memory	4GB RAM
Storage	32GB Memory (expandable)
Wi-Fi	802.11ac
Camera	4 MP camera with MJPEG
Software	Ubuntu OS 16.04.3 LTS, Intel® OpenVINO™ toolkit, ROS Kinetic

Fonte: AWS**Tabela 13.2.2** - Configuração do carro autônomo da AWS

Drive Battery	7.4V/1100mAh lithium polymer
Compute Battery	13600mAh USB-C PD
Ports	4x USB-A, 1x USB-C, 1x Micro-USB, 1x HDMI
Sensors	Integrated accelerometer and gyroscope

Fonte: AWS**Figura 13.2.4** - Modelo do AWS *DeepRacer* Evo, que tem câmera estéreo e um sensor Lidar**Fonte:** AWS

Figura 13.2.5 - Visão lateral do SDC da AWS com sensor lidar



Fonte: AWS

Figura 13.2.6 - Visão em perspectiva do AWS DeepRacer Evo com câmeras Estéreo e sensor Lidar



Fonte: AWS

Figura 13.2.7 - Inclusão de mais uma câmera frontal e um sensor Lidar



Fonte: AWS

Figura 13.2.8 - Inclusão do Sensor Lidar no AWS DeepRacer



Fonte: AWS

Tabela 13.2.3 - Configurações do AWS DeepRacer EVO

Camera	4 MP camera with MJPEG
Lidar Configurable Scan Rate	10 Hz
Lidar Distance Data Measurement Rate	8000 times/s
Lidar Laser Scanning Range	360 degrees
Integrated Sensors	Accelerometer and Gyroscope

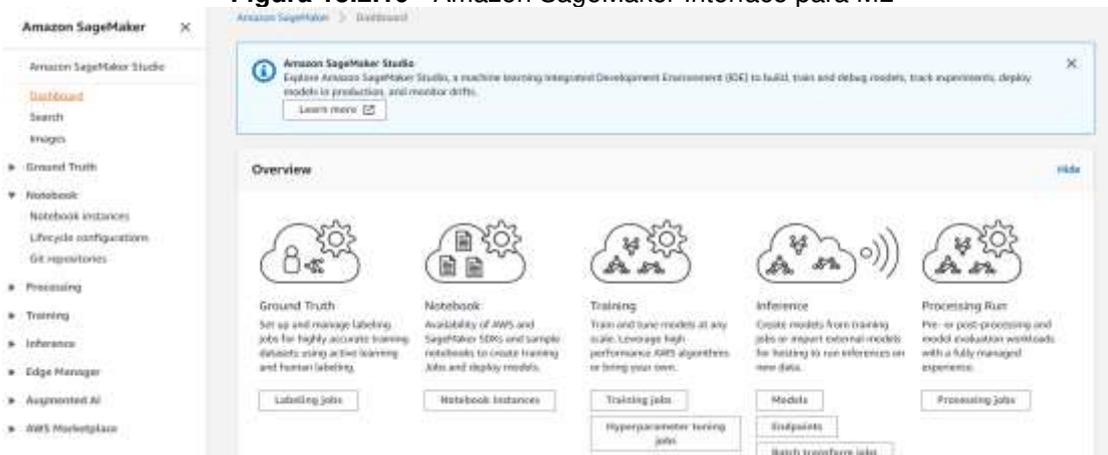
Fonte: AWS

Figura 13.2.9 - Interface Inicial da AWS DeepRacer



Fonte: AWS

Figura 13.2.10 - Amazon SageMaker Interface para ML



Fonte: AWS

Figura 13.2.11 - Amazon SageMaker Tela Inicial

The image shows the Amazon SageMaker console interface. On the left is a navigation sidebar with the following items: Amazon SageMaker Studio, Dashboard, Search, Images, Ground Truth, Notebook, Processing, Training, Inference, Edge Manager, Augmented AI, and AWS Marketplace. The main content area has a dark header with the text 'MACHINE LEARNING' and 'Amazon SageMaker Build, train, and deploy machine learning models at scale'. Below this is a sub-header 'The quickest and easiest way to get ML models from idea to production.' To the right of the header is a 'Get started' section with a 'SageMaker Studio' button. Below the header is a 'How it works' section with three icons labeled 'Label', 'Build', and 'Train'. To the right of the 'How it works' section is a 'Pricing (US)' section with text explaining the pay-as-you-go model and a 'Learn more' link.

Amazon SageMaker

Amazon SageMaker Studio

Dashboard

Search

Images

Ground Truth

Notebook

Processing

Training

Inference

Edge Manager

Augmented AI

AWS Marketplace

MACHINE LEARNING

Amazon SageMaker

Build, train, and deploy machine learning models at scale

The quickest and easiest way to get ML models from idea to production.

Get started

Explore SageMaker Studio, a machine learning Integrated Development Environment (IDE) for building, training, and debugging models, tracking experiments, deploying models, and monitoring their performance.

[SageMaker Studio](#)

How it works

Label

Build

Train

Pricing (US)

With Amazon SageMaker, you pay only for what you use. Appending, training, and hosting is billed by the second, with no minimum fees and no upfront commitments.

[Learn more](#)

Fonte: AWS

14 ANEXO II – FOTOS DA SIMULAÇÃO DOS MODELOS NA AWS DEEPRACER

14.1 TESTANDO O MODELO

Figura 14.1.1 - - ScreenShot da Evaluation do 4º do Modelo



Fonte: Autor

Figura 14.1.2 - ScreenShot da Evaluation da 5ª versão do modelo em uma outra pista



Fonte: Autor

15 ANEXO III – SEQUÊNCIA DE FUNCIONAMENTO DA ENTREGA DE PIZZA POR CARRO AUTÔNOMO

15.1 DELIVERY AUTÔNOMO

Figura 15.1.1 - Carro autônomos fica esperando na área de abastecimento da pizzaria



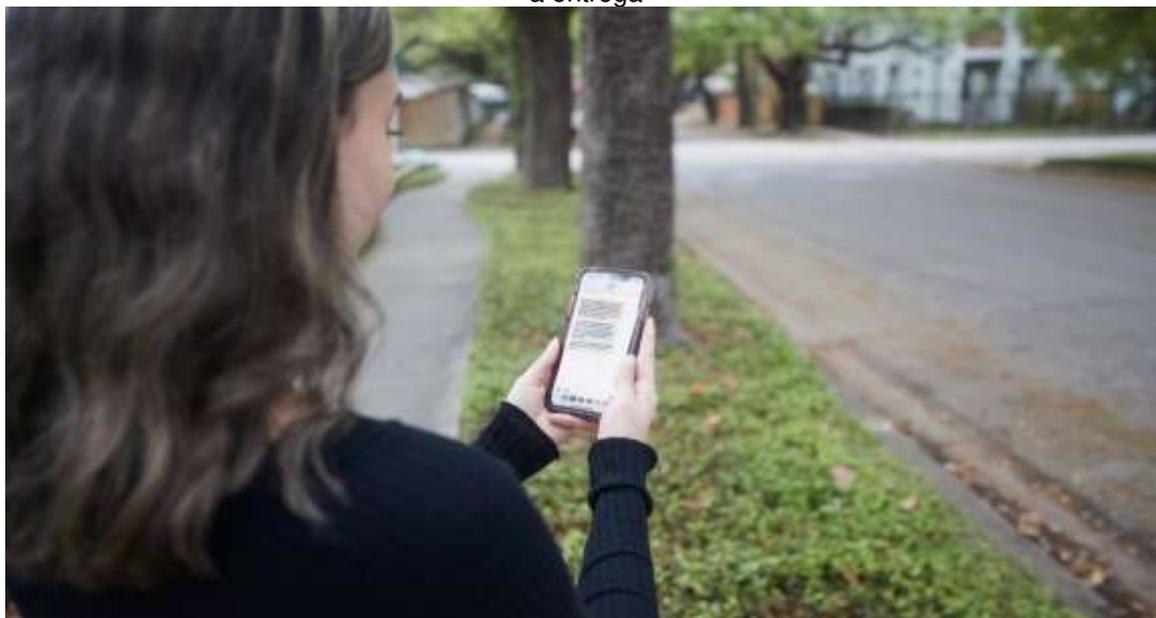
Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

Figura 15.1.2 - Funcionário acomoda a pizza no compartimento de produto do carro autônomo



Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

Figura 15.1.3 - Cliente recebe uma notificação que seu pedido já foi carregado e já saiu para a fazer a entrega



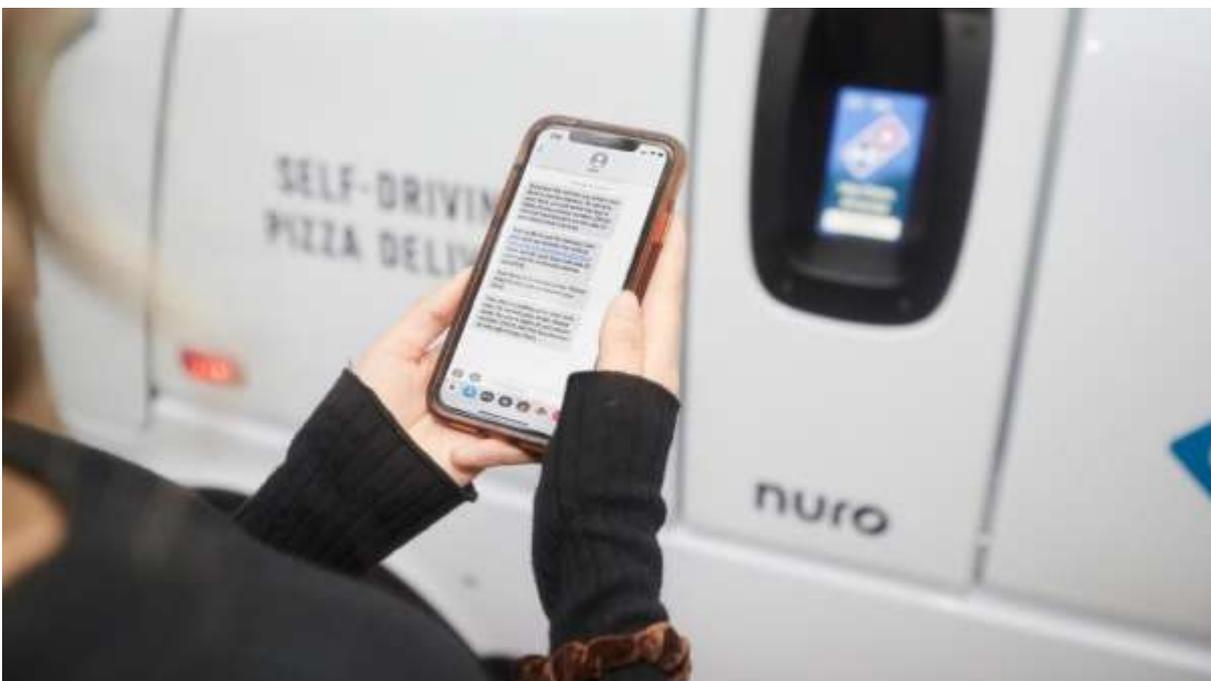
Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

Figura 15.1.4 - A Empresa consegue acompanhar o trajeto do carro autônomo em tempo real



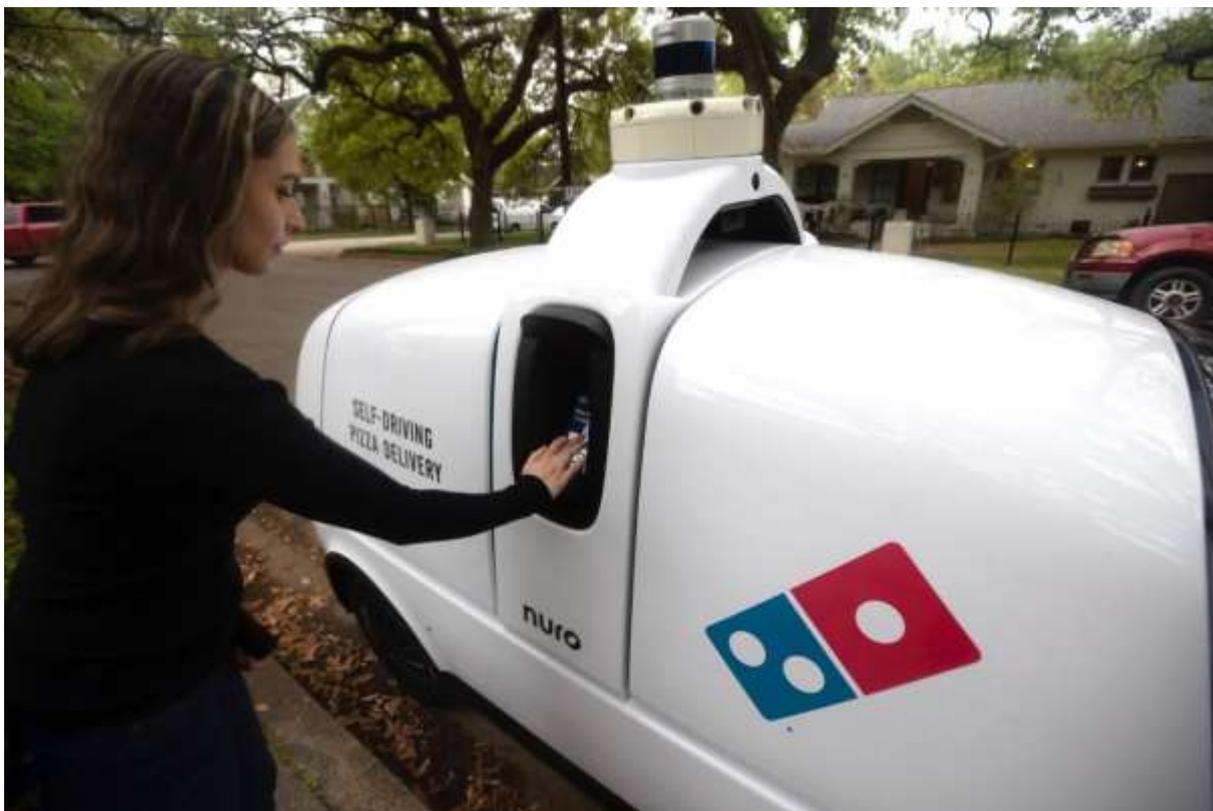
Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

Figura 15.1.5 - Ao chegar o SDC, é enviada uma notificação ao cliente sobre a chegada do produto



Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

Figura 15.1.6 - Cliente entra com seu código único do seu pedido para a retirada do seu produto



Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

Figura 15.1.7 - Após o cliente entrar com seu código, o SDC libera para o cliente a retirada do seu produto



Fonte: https://www.motorauthority.com/news/1131910_domino-s-launches-autonomous-pizza-delivery-with-self-driving-robot-car

16 ANEXO IV – SIMULADORES DE CARROS AUTÔNOMOS

A seguir serão apresentados os melhores simuladores virtuais para carros autônomos. Quase todos são *open sources* e estão disponíveis para uso direto no link da legenda da imagem.

16.1 CARLA

Figura 16.1.1 - Página principal do Simulador Carla



Fonte: <https://carla.org>

Figura 16.1.2 - Empresas envolvidas no desenvolvimento do simulador Carla



Fonte: <https://carla.org/>

16.2 UDACITY SELF-DRIVE CAR

Figura 16.2.1 - Screen Shot do Simulador da Udacity



Fonte: <https://github.com/udacity/self-driving-car-sim>

16.3 NVIDIA DRIVE SIM

Figura 16.3.1 - Simulador da Nvidia



Fonte: <https://developer.nvidia.com/drive/drive-sim>

16.4 AIR SIM

Figura 16.4.1 - AirSim -simulador open source da Microsoft que pode ser usado até para simulação de drones



Fonte: <https://microsoft.github.io/AirSim/>

17 APENDICE A – CÓDIGOS

17.1 CODIGOS PARA PROCESSAMENTO DE IMAGEM

Figura 17.1.1 - Exemplo da aplicação do método cvtColor()

```
# importing cv2
import cv2

# path
path = r'C:\Users\Administrator\Desktop\geeks.png'

# Reading an image in default mode
src = cv2.imread(path)

# Window name in which image is displayed
window_name = 'Image'

# Using cv2.cvtColor() method
# Using cv2.COLOR_BGR2GRAY color space
# conversion code
image = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY )

# Displaying the image
cv2.imshow(window_name, image)
```

Fonte: Autor

Figura 17.1.2 - Exemplo de aplicação do filtro Gaussiano de suavização

```
import cv2
import numpy

# read image
src = cv2.imread('/home/img/python.png', cv2.IMREAD_UNCHANGED)

# apply gaussian blur on src image
dst = cv2.GaussianBlur(src,(5,5),cv2.BORDER_DEFAULT)

# display input and output image
cv2.imshow("Gaussian Smoothing",numpy.hstack((src, dst)))
```

Fonte: Autor

Figura 17.1.3 - Aplicação da função threshold na imagem em gradientes de cinza para obter uma imagem binária

```
otsu_threshold, image_result = cv2.threshold(
    image, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU,
)
print("Obtained threshold: ", otsu_threshold)
```

Fonte: Autor

Figura 17.1.4 - Trecho do código para obtenção do dataset

```
datadir = 'track'
columns = ['center', 'left', 'right', 'steering', 'throttle', 'reverse', 'speed']
data = pd.read_csv(os.path.join(datadir, 'driving_log.csv'), names = columns)
pd.set_option('display.max_colwidth', -1)
data.head()
```

Fonte: Autor

Figura 17.1.5 - Desenvolvimento da Função de recompensa

```
def reward_function(params):
    if not params["all_wheels_on_track"]:
        reward = -1
    else if params["progress"] == 1 :
        reward = 10
    return reward
```

Fonte: Autor

Figura 17.1.6 - Versão da Função de recompensa do modelo RL

```
def reward_function(params):
    # Leitura dos parametros de entrada
    all_wheels_on_track = params['all_wheels_on_track']
    distance_from_center = params['distance_from_center']
    track_width = params['track_width']
    steering = abs(params['steering_angle']) # Obtenção do angulo absoluto da direção

    # Calculando 4 marcadores em relação a distancia ao linha do centro
    marker_1 = 0.2 * track_width
    marker_2 = 0.3 * track_width
    marker_3 = 0.4 * track_width
    marker_4 = 0.45 * track_width
    marker_5 = 0.5 * track_width + 0.05

    # Dando uma recompensa alta se todas as rodas não sairem da trajetória
    #
    if all_wheels_on_track:

        if distance_from_center <= marker_1:
            reward = 1
        elif distance_from_center <= marker_2:
            reward = 0.95
        elif distance_from_center <= marker_3:
            reward = 0.7
        elif distance_from_center <= marker_4:
            reward = 0.3
        elif distance_from_center <= marker_5:
            reward = 0.1
        else:
            reward = 1e-3 # saiu da trajetória
    else:
        # dar uma recompensa baixa por padrão
        reward = 1e-3

    # Penalidade de direção, muda o número conforme a ação
    abs_steering_threshold = 20

    # Penaliza caso o SDC disterça muito
    if steering > abs_steering_threshold:
        reward *= 0.8

    # Retorno do valor é do tipo float

    return float(reward)
```

Fonte: Autor