

CENTRO UNIVERSITÁRIO DO PARÁ - CESUPA
ESCOLA DE NEGÓCIOS, TECNOLOGIA E INOVAÇÃO - ARGO
CURSO DE ENGENHARIA DA COMPUTAÇÃO

DANIEL MASAYUKI ARANHA MATSUNAGA
MARCELO MELO ALAMAR DE SOUSA
MARCUS MATHEUS PEREIRA GOMES
WALLACE RICARDO DA SILVA

SISTEMA DE TRÁFEGO INTELIGENTE: SITI

BELÉM
2023

DANIEL MASAYUKI ARANHA MATSUNAGA
MARCELO MELO ALAMAR DE SOUSA
MARCUS MATHEUS PEREIRA GOMES
WALLACE RICARDO DA SILVA

SISTEMA DE TRÁFEGO INTELIGENTE: SiTI

Trabalho de conclusão de curso apresentado à Escola de Negócios, Tecnologia e Inovação do Centro Universitário do Estado do Pará como requisito para obtenção do título de Engenheiro da Computação na modalidade PRODUTO.

Orientador(a): Me. Pedro Henrique Sales Giroto

BELÉM
2023

DANIEL MASAYUKI ARANHA MATSUNAGA
MARCELO MELO ALAMAR DE SOUSA
MARCUS MATHEUS PEREIRA GOMES
WALLACE RICARDO DA SILVA

SISTEMA DE TRÁFEGO INTELIGENTE: SiTI

Trabalho de conclusão de curso apresentado à Escola de Negócios, Tecnologia e Inovação do Centro Universitário do Estado do Pará como requisito para obtenção do título de Engenheiro da Computação na modalidade PRODUTO.

Data da aprovação: 6/12/23

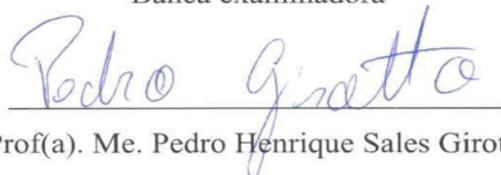
Nota final aluno(a) I: _____

Nota final aluno(a) II: _____

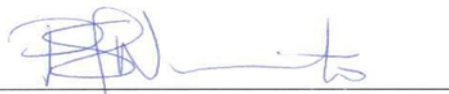
Nota final aluno(a) III: _____

Nota final aluno(a) IV: _____

Banca examinadora



Prof(a). Me. Pedro Henrique Sales Giroto
Orientador(a) e Presidente da banca



Prof(a). Ma. Polyana Santos Fonseca Nascimento
Examinador(a) interno(a)



Prof(a). Dr. Isaac Souza Elgabrly
Examinador(a) interno(a)

Dados Internacionais de Catalogação-na-publicação (CIP)

Biblioteca do CESUPA, Belém – PA

Matsunaga, Daniel Masayuki Aranha.

Sistema de tráfego inteligente: SiTI / Daniel Masayuki Aranha
Matsunaga, Marcelo Melo Alamar de Sousa, Marcus Matheus Pereira
Gomes, Wallace Ricardo da Silva; orientador Pedro Henrique Sales
Giroto. — 2023.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia
de Computação) – Centro Universitário do Estado do Pará, Belém,
2023.

1. Inteligência artificial. 2. Mobilidade urbana. 3. Software –
Desenvolvimento. I. Sousa, Marcelo Melo Alamar. II. Gomes, Marcus
Matheus Pereira. III. Silva, Wallace Ricardo da. IV. Giroto, Pedro
Henrique Sales, orient. V. Título.

CDD 23^a ed.

006.3



Dedico este trabalho a minha avó Jandira Valente por todo apoio que imbuí em mim durante toda sua vida terrena.

AGRADECIMENTOS

Agradeço a minha família por sempre me apoiar em tudo. A minha namorada que diariamente me incentiva e me acompanha. Um agradecimento especial para meus apoiadores Ubiratan da Veiga Pereira, Iracema da Veiga Pereira e Moema da Veiga Pereira, na qual se dispuseram a me ajudar durante todos esses momentos decisivos em minha vida acadêmica. (Marcus Matheus)

Este trabalho de conclusão de curso não teria sido possível sem o apoio e incentivo de muitas pessoas. Gostaria de expressar minha profunda gratidão a todas que me ajudaram a alcançar este objetivo. Gostaria de agradecer aos meus pais pelo amor, apoio e incentivo incondicionais em todos os momentos da minha vida. Sem o amor e o apoio deles, eu não teria chegado até aqui e não seria a pessoa que sou hoje. Seus sacrifícios e dedicação possibilitaram a minha formação acadêmica e por isso estou profundamente grato. (Daniel Masayuki)

Para minha família, agradeço a cada um de vocês por serem a minha base, por acreditarem em mim quando a autoconfiança vacilava e por celebrarem cada passo dado rumo a esta conquista. Este diploma não é apenas meu, é nosso, pois reflete o esforço coletivo, o amor e a persistência que caracterizam a nossa família. (Marcelo Melo)

Gostaria de expressar minha profunda gratidão a duas presenças especiais que tornaram esta jornada acadêmica ainda mais significativa. À minha querida mãe, cujo apoio incansável e incentivo constante foram os alicerces que me sustentaram durante todo o percurso do TCC. E também não posso deixar de agradecer à minha adorável gata de estimação, companheira silenciosa e reconfortante. Ambas são peças fundamentais neste capítulo da minha vida, e este trabalho é dedicado a vocês, que permitiram transformar desafios em conquistas. (Wallace Ricardo)

LISTA DE SIGLAS

CNDL - Confederação Nacional de Dirigentes Lojistas

CNN - Redes Neurais Convolucionais

IA - Inteligência artificial

IBGE - Instituto Brasileiro de Geografia e Estatística

NMS - *Non Maximum Suppression*

OCDE - Organização para a Cooperação e Desenvolvimento Econômico

OpenCV - Open Source Computer Vision Library

RGB - *Red, green, blue*

SAC - Sistema de atendimento ao cliente

SPC - Serviço de proteção ao crédito

YOLO - *You Only Look Once*

SUMÁRIO

RESUMO	9
1 CONTEXTUALIZAÇÃO	11
1.1 Introdução	11
1.2 Problema	12
1.3 Justificativa	13
1.4 Objetivos	14
1.5 Estrutura do trabalho	15
2 DESENVOLVIMENTO DO PRODUTO	16
2.1 Análise de viabilidade	16
2.2 Prototipação	18
3 RESULTADOS E DISCUSSÃO	26
4 CONSIDERAÇÕES FINAIS	31
5 REFERÊNCIAS BIBLIOGRÁFICAS	32

RESUMO

Mobilidade urbana é o termo utilizado para se referir ao deslocamento de pessoas pelo espaço urbano, ou seja, se trata do ir e vir das pessoas nas cidades. A partir disso, a integração de tecnologias avançadas na mobilidade urbana é fundamental para melhorar a qualidade de vida e enfrentar os desafios da urbanização. A precariedade das redes de transporte, advindas da falta de um planejamento adequado, somadas ao descaso governamental são causas responsáveis pela má qualidade de vida no tráfego, causando congestionamentos e acidentes. O objetivo deste trabalho é melhorar os problemas de trânsito utilizando da inteligência artificial para desenvolver um sistema de tráfego inteligente. Para isto foi desenvolvido um software de reconhecimento de imagem em tempo real utilizando as tecnologias do YOLO e OpenCV, o mesmo foi capaz de reconhecer, diferenciar e contabilizar os diferentes tipos de veículos. Os resultados obtidos foram um protótipo capaz de extrair dados com alta precisão, o que por sua vez permite a aplicação dos valores obtidos a serem aplicados em cálculos para o controle dos semáforos.

Palavras-chave: Mobilidade urbana; Inteligência artificial; Tráfego inteligente; Linguagem Python; YOLO.

ABSTRACT

Urban mobility is the term used to refer to the movement of people in urban spaces, in other words, it involves the coming and going of people in cities. Based on this, the integration of advanced technologies in urban mobility is crucial to improve the quality of life and address the challenges of urbanization. The inadequacy of transportation networks, stemming from a lack of proper planning, combined with governmental neglect, is a cause of poor quality of life in traffic, leading to congestion and accidents. The aim of this work is to address traffic problems by utilizing artificial intelligence to develop an intelligent traffic system. To achieve this, a real-time image recognition software was developed using YOLO and OpenCV technologies. It was able to recognize, differentiate, and count different types of vehicles. The results obtained include a prototype capable of extracting data with high precision, allowing the application of these values in calculations for traffic light control.

Keywords: Urban mobility; Artificial intelligence; Intelligent traffic; Python language; YOLO.

1 CONTEXTUALIZAÇÃO

1.1 Introdução

Segundo Mumford (1982) o surgimento das primeiras cidades se deu a milhares de anos atrás durante o período neolítico, e junto a isso começou o processo de formação da ideia de urbanização. A urbanização se trata do crescimento das cidades tanto em população quanto em extensão territorial. Ela é processo em que o espaço rural se transforma em espaço urbano, com a consequente migração populacional rural para o âmbito urbano. Esse processo conforme o conhecemos hoje se iniciou no século XVIII durante a revolução industrial, diferentemente do Brasil, cuja urbanização ocorreu apenas na segunda metade do século XX (SOUSA, 2021). A Figura 1 mostra a estimativa da população brasileira em 2021 separadas por regiões, e estima-se que 85% da população seja de natureza urbana (ALVES, 2022).

Figura 1 – Estimativas da população brasileira em 2021

ESTIMATIVAS DA POPULAÇÃO RESIDENTE NO BRASIL E UNIDADES DA FEDERAÇÃO COM DATA DE REFERÊNCIA EM 1º DE JULHO DE 2021	
BRASIL E UNIDADES DA FEDERAÇÃO	
Brasil	213.317.639
Região Norte	18.906.962
Rondônia	1.815.278
Acre	906.876
Amazonas	4.269.995
Roraima	652.713
Pará	8.777.124
Amapá	877.613
Tocantins	1.607.363
Região Nordeste	57.667.842
Maranhão	7.153.262
Piauí	3.289.290 ⁽¹⁾
Ceará	9.240.580 ⁽¹⁾
Rio Grande do Norte	3.560.903
Paraíba	4.059.905
Pernambuco	9.674.793 ⁽²⁾
Alagoas	3.365.351 ⁽²⁾
Sergipe	2.338.474 ⁽³⁾
Bahia	14.985.284 ⁽³⁾
Região Sudeste	89.632.912
Minas Gerais	21.411.923
Espírito Santo	4.108.508
Rio de Janeiro	17.463.349
São Paulo	46.649.132
Região Sul	30.402.587
Paraná	11.597.484
Santa Catarina	7.338.473
Rio Grande do Sul	11.466.630
Região Centro-Oeste	16.707.336
Mato Grosso do Sul	2.839.188
Mato Grosso	3.567.234
Goiás	7.206.589 ⁽⁴⁾
Distrito Federal	3.094.325 ⁽⁴⁾

Fonte: IBGE, estimativas da população brasileira em primeiro de julho, 2021

A Revolução Industrial foi um processo de grandes transformações tecnológicas, sociais e econômicas que levou à substituição das ferramentas pelas máquinas, e do modo de

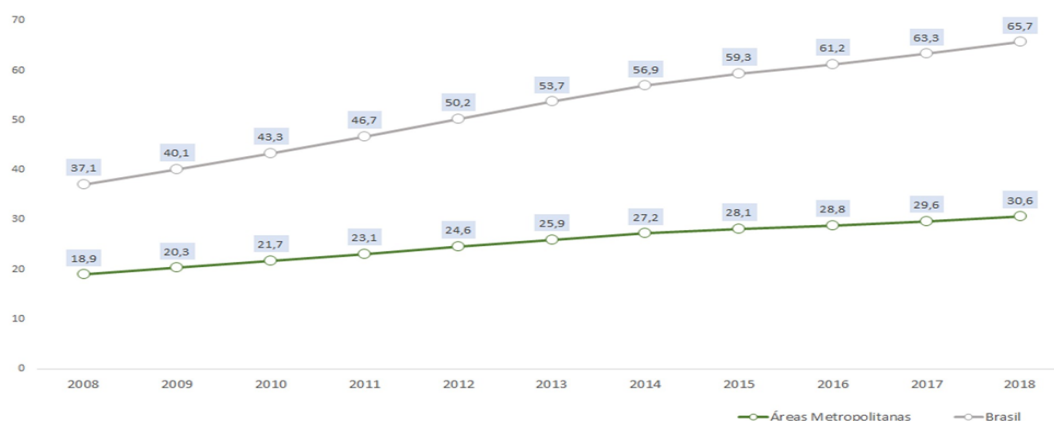
produção doméstico pelo sistema fabril (BEZERRA, 2023). A quarta revolução é a causa de impactos econômicos, éticos, sociais e ambientais por meio de novas tecnologias como inteligência artificial, internet das coisas e robótica (MAGALHÃES, 2018). Essas tecnologias resultam na melhoria de qualidade de vida nas áreas urbanas, otimizar a infraestrutura e a gestão urbana, de modo que promova a sustentabilidade, assim como sistemas de energia solar, gestão de resíduos inteligente, redes de comunicação avançadas, e o foco desse trabalho que é sistemas de transporte inteligentes (CORTESE, 2019).

Com o contínuo processo de urbanização nas grandes cidades, a capacidade de um indivíduo de se locomover de maneira rápida e eficiente nas grandes cidades tem se tornado um desafio complexo a ser resolvido. Segundo o art.4º da lei nº 12.587, de 3 de janeiro de 2012, à qual conceitua que mobilidade urbana como a condição em que se realizam os deslocamentos de pessoas e cargas no espaço urbano (BRASIL, 2012). A mobilidade urbana não se trata apenas de se locomover e sim se locomover com fluidez, qualidade e praticidade, seja mediante veículos motorizados como carros e motos, ou transportes públicos como ônibus e trens, e veículos não motorizados como as bicicletas. Para melhorar a qualidade de vida das pessoas de forma sustentável é necessário interesse por parte da população e comprometimento do poder público para novas soluções (ARAÚJO, 2011).

1.2 Problema

Segundo os dados da OCDE (2023), o Brasil tem os piores índices de qualidade de estradas, congestionamento e índice de mortalidade alto (16 a cada 100 mil pessoas), fazendo com que seja o 2º no ranking de piores países do mundo, se devendo a uma combinação de fatores, como a falta de planejamento urbano adequado, infraestrutura deficiente e alta densidade populacional em áreas metropolitanas. Isso resulta em congestionamentos diários, perda de tempo e aumento da poluição do ar, sonora e visual devido ao aumento de veículos nos últimos anos (Figura 2).

Figura 2 - Frota de automóveis no Brasil e Áreas Metropolitanas - 2008 a 2018



Fonte: Elaborado pelo Observatório das Metrôpoles com dados do Denatran, 2019

O incentivo político do país no sistema de transporte rodoviário desencadeou no Brasil a cultura do automóvel, fazendo com que os veículos deixem de ser um simples meio de transporte para se tornar um símbolo de status social, representando a independência econômica, sucesso pessoal e liberdade, alimentada pela ideia de que através dos veículos pode ir para onde se deseja e da maneira que se deseja, sendo um dos fatores responsáveis pelos problemas enfrentados hoje pela mobilidade urbana brasileira (SCHNEIDER, 2021). Tendo em vista esse ponto, a população sofreu com a falta de segurança devido ao alto número de veículos nas vias, congestionamentos e um trânsito caótico na maioria das grandes cidades brasileiras.

A qualidade de vida no trânsito está diretamente relacionada aos sistemas de transporte, principalmente o coletivo, para disponibilizar acessibilidade e facilidade na mobilidade para garantir autonomia ao cidadão (ARAÚJO, 2011). Com o passar dos anos as cidades brasileiras estão cada vez mais dando foco para automóveis, prejudicando a qualidade de vida com o congestionamento de veículos e falta de acessibilidade na mobilidade.

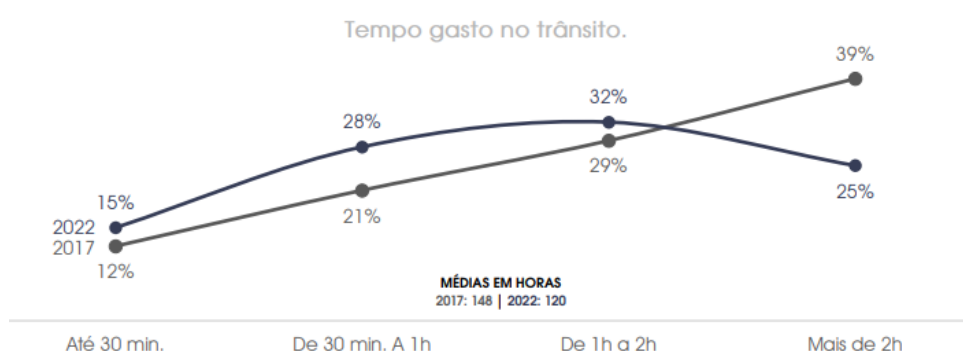
Com os fatos apresentados, quais medidas podem ser tomadas para reduzir o congestionamento no tráfego? E qual a necessidade de tomar essas medidas?

1.3 Justificativa

Uma cidade com um sistema de tráfego inteligente, interligado e conectado traz muitos benefícios em curto, médio e longo prazo tanto para a população, proporcionando mais conforto e qualidade de locomoção, quanto para abrir portas para outras tendências de uma cidade inteligente.

O benefício mais óbvio de um sistema de tráfego inteligente é a melhoria da mobilidade e, conseqüentemente, a qualidade de vida, com redução do tempo no trânsito e a diminuição do estresse relacionado ao tráfego, o qual impacta diretamente no dia a dia e bem-estar dos cidadãos. Visto que uma pesquisa realizada pela CNDL e pelo SPC Brasil em parceria com o Sebrae, a maioria dos brasileiros que moram nos grandes centros urbanos passam cerca de 2h do seu dia no trânsito, o que pode chegar a cerca de 21 dias no ano (BARBOSA, 2022). Como mostrado na Figura 3.

Figura 3 - Estatística de tempo gasto no trânsito feito pela Pesquisa Mobilidade Urbana 2022



Fonte: Mobilidade Urbana, 2022

A implementação de um sistema como esse pode reduzir o número de acidentes no trânsito, com auxílio de diversas implementações que auxiliem no fluxo de veículos e na segurança no trânsito, além disso, uma gestão mais eficiente do tráfego pode reduzir os custos associados ao transporte, tanto para o governo quanto para os cidadãos, com menor tempo gasto no trânsito haverá maior economia de combustível e menor desgaste dos veículos.

1.4 Objetivos

1.4.1 Geral

Criar um sistema de tráfego inteligente por meio da utilização de inteligência artificial com reconhecimento por câmera.

1.4.2 Específicos

- Coletar e analisar dados de tráfego em tempo real.
- Desenvolver algoritmos de inteligência artificial para prever e evitar congestionamentos.
- Implementar sistemas de sinalização inteligente para melhorar o fluxo de tráfego.
- Integrar sistemas de monitoramento de segurança no trânsito.

1.5 Estrutura do trabalho

Neste trabalho no capítulo um será apresentado uma introdução e contextualização sobre urbanização e mobilidade urbana, a problemática que se pretende solucionar com esse produto, a justificativa e os objetivos do trabalho. No segundo capítulo será apresentado o desenvolvimento do produto a partir da análise de viabilidade com o mercado, competências técnicas e as limitações, em seguida vem a prototipação com as tecnologias utilizadas, funcionalidades do produto, os testes realizados e o método de venda. No terceiro capítulo virá os resultados obtidos e a discussão sobre os mesmos, e no quarto capítulo virá as considerações finais do projeto.

2 DESENVOLVIMENTO DO PRODUTO

Foram feitos estudos e pesquisas para verificar a competitividade de mercado e analisar se é viável uma comercialização do projeto.

2.1 Análise de viabilidade

2.1.1 Mercado e Público-alvo

Gerir um trânsito de qualidade é uma dificuldade muito encontrada atualmente, por conta disso nosso produto terá como atuar no mercado de tráfego inteligente com a comercialização do software para órgãos governamentais que organizam o trânsito de grandes cidades, facilitando assim a sua implementação, será feita uma prestação de serviços para manutenção do mesmo realizada por uma equipe responsável.

2.1.2 Competências Técnicas

Competência é definida como a capacidade que um indivíduo possui de expressar um juízo de valor sobre algo a respeito de que é versado, a partir disso podemos analisar a competência como um conjunto de capacidades humanas que justificam um alto desempenho (FLEURY, 2009). Entre outras palavras, a competência é percebida como um estoque de recursos que o indivíduo detém, ela não é um estado, e não se reduz a um conhecimento específico. Não somente as competências técnicas são importantes para que o engenheiro da computação tenha uma carreira de sucesso, mas também existe a necessidade do desenvolvimento de competências não técnicas, como aquelas relacionadas ao trabalho em grupos, comunicação, resiliência, entre outras, denominadas *soft skills* (LIMA, 2018).

Para o desenvolvimento desse software é necessário que os programadores tenham conhecimento sólido em lógica de programação para entender o fluxo de trabalho por trás da programação. Um ponto importante é um conhecimento consistente nos métodos de inteligência artificial, pois os métodos para manipular os dados são de conhecimentos específicos da área.

Como base principal do código será utilizado a programação em Python para manuseio da biblioteca YOLO para que tudo seja integrado na mesma linguagem.

2.1.3 Análise financeira

Para uma melhor compreensão e organização da nossa análise financeira foi realizado um canvas para ter uma melhor visualização das nossas ferramentas e custos.

Figura 4 - Modelo de Negócios

Canvas SiTI



Fonte: Autores, 2023

Não foi realizada nenhuma parceria com outras empresas ou obtida qualquer ajuda externa, sendo o produto resultado de um trabalho autoral e independente, utilizando exclusivamente ferramentas *open source* já existentes. Ao longo do projeto, empregamos diversos recursos, desde a criação de códigos para o desenvolvimento do algoritmo até reuniões semanais para discutir o progresso, visando otimizar o desempenho da equipe e do produto.

O foco dos meios de comunicação e divulgação está no marketing e na publicidade, com estudos e análises de mercado para direcionar eficientemente propagandas e divulgações, identificando públicos-alvo. Os custos do produto serão alocados para a implementação e manutenção do sistema. Isso se deve à necessidade de um servidor em nuvem para armazenar os vídeos gravados, com uma estimativa de despesas de R\$750 por mês para manter o servidor ativo para apenas uma única câmera. Essa estimativa aumentará proporcionalmente com a expansão do projeto.

O produto pode ser adquirido por meio de licitações ou contratações de serviços, direcionado a órgãos governamentais como público-alvo, com suporte fornecido mediante um SAC para garantir contato direto e eficiente. A proposta principal do SiTI é o desenvolvimento de uma Inteligência Artificial para o controle autônomo do trânsito, visando a redução do congestionamento e aprimoramento do monitoramento do tráfego.

2.1.4 Limitações

A tecnologia utilizada continua em processo de desenvolvimento de sua rede neural, então ainda necessita de bastante dados no seu *dataset*, portanto é necessário um contínuo treinamento para alcançar um alto grau de precisão para evitar possíveis falhas no reconhecimento das classes. Por se tratar de uma tecnologia de reconhecimento de imagens há algumas limitações enfrentadas que é a qualidade de imagens em algumas câmeras.

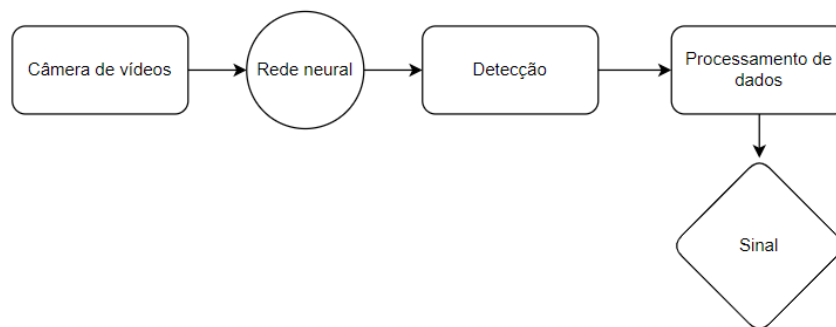
2.2 Prototipação

2.2.1 Requisitos

Para uma compreensão do fluxo do projeto foi desenvolvido um diagrama de processos de esclarecimento.

2.2.1.1 Infográfico

Figura 5 - Infográfico do Projeto



Fonte: Autores, 2023

2.2.2 Funcionalidades do Produto

Mediante uma captura de vídeos é realizado uma detecção das imagens como uma ferramenta de redes neurais já treinadas previamente que estão separadas por classes e conforme os dados retornados é realizada uma contagem dos veículos na via, segundo a quantidade é realizado um cálculo que retorna um valor mais adequado para os tempos de cada cor do semáforo para que com isso possa ser otimizado o fluxo de veículos.

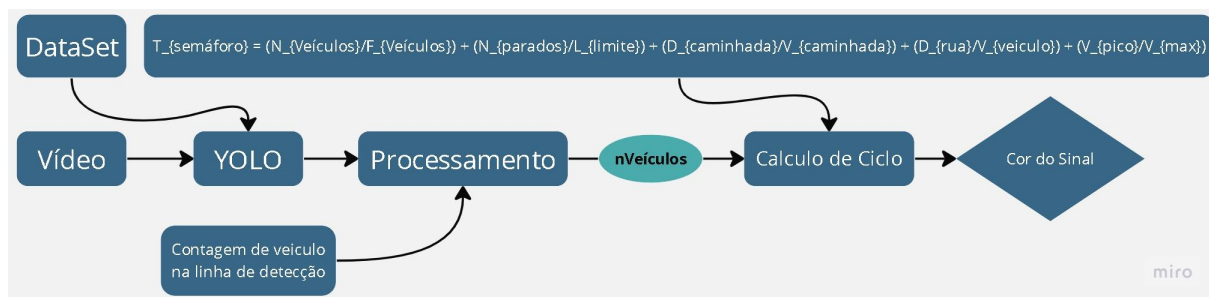
2.2.3 Análise de Dados/Uso de Arquivos de Entrada e Saída

- Capturar vídeos da pista de tráfego utilizando câmeras de monitoramento.
- Aplicar algoritmos de detecção de objetos através do YOLO.
 - Etapa onde realmente ocorre a detecção dos objetos aplicando o método *You Only Live Once*.
- Realizar a contagem de veículos com base nas informações de rastreamento, utilizando métodos com contagem por detecção de entrada/saída de áreas específicas.
 - Na etapa de processamento ocorre a coleta de dados a partir da detecção realizada pelo YOLO onde é extraído valores a partir de uma área de detecção de interesse, que conforme a passagem dos objetos nessa área é alterado o valor do veículo, realizando assim a contagem dos automóveis.
- Para a temporização dos sinais de trânsito é realizado um cálculo base para saber o valor ideal para ser contado pelo controlador de sinal.
 - $T_{\text{semáforo}}$: Tempo total do semáforo.
 - $N_{\text{veículos}}$: Número de veículos.
 - $F_{\text{veículos}}$: Frequência de veículos (presumivelmente, o número de carros por unidade de tempo).
 - N_{parados} : Número de veículos parados.
 - L_{limite} : Limite de carros que podem ser parados
 - $D_{\text{caminhada}}$: Distância de caminhada (distância que os pedestres precisam percorrer).
 - $V_{\text{caminhada}}$: Velocidade de caminhada (velocidade média na qual os pedestres estão andando).
 - D_{rua} : Largura da rua (largura da rua que os carros escoaram).
 - $V_{\text{veículo}}$: Velocidade do veículo (a velocidade média na qual os carros estão se movendo).
 - V_{pico} : Velocidade de pico (a velocidade máxima atingida pelos carros).
 - V_{max} : Velocidade máxima permitida.
- Enviar os dados de contagem de veículos para um cálculo de estimativa de tempo dos sinais de trânsito.

Tendo toda a explicação detalhada pode-se definir os parâmetros, como os *inputs* são declaradas qualquer imagem capturada pela câmera com auxílio da biblioteca do openCV

com o YOLO. Em output já fica definido um sinal contendo o estado que deve se encontrar o semáforo de trânsito, como visto na figura 6.

Figura 6 - Diagrama de processos do SiTI



Fonte: Autores, 2023

2.2.4 Tecnologias Utilizadas

2.2.4.1 Python

Python é uma linguagem de programação de alto nível ao qual foi utilizada para este projeto, é uma linguagem multiplataforma versátil e amigável para iniciantes, simples e direta com uma sintaxe clara e legível, fácil de se aprender e pode ser aplicada a diversas áreas incluindo na de inteligência artificial, possui diversas bibliotecas que facilita na implementação de aplicações complexas sem haver a necessidade de codificar do zero (SANTANA, 2023).

2.2.4.2 OpenCV

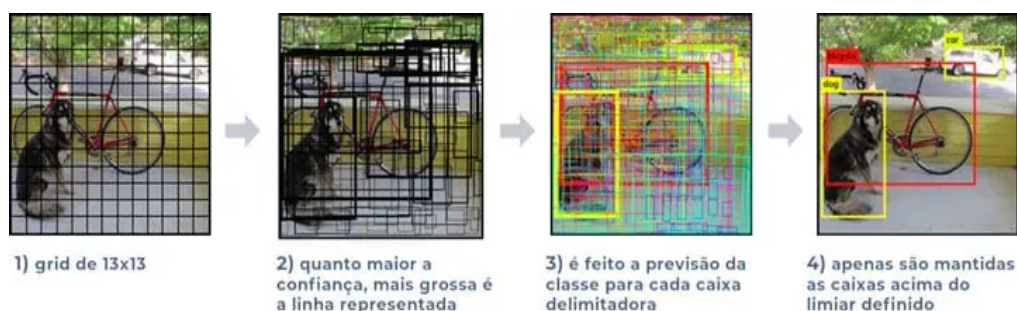
OpenCV é uma biblioteca de código aberto inicialmente escrita com a linguagem C e C++ contendo diversos algoritmos de visão computacional, detecção de objetos, segmentação de imagem e rastreamento de movimento. O OpenCV possui uma estrutura modular, que nada mais é que ele possui diversos pacotes de bibliotecas compartilhadas e estáticas. Possui a capacidade de acessar diretamente dispositivos de entrada, como câmeras e processar imagem e vídeo de diversos formatos em tempo real (MARQUES, 2018). OpenCV traz ferramentas de visão computacional que torna mais fácil e rápido desenvolver aplicações eficientes e sofisticadas (BRADSKI, 2008).

2.2.4.3 YOLO

O YOLO utiliza um sistema de rede neural profunda e será possível realizar uma análise das imagens mais rápida e eficiente, pois divide a imagem em uma grade e, para cada

célula da grade, prevê caixas delimitadoras (*bounding boxes*) que cercam os objetos detectados e as classes a que esses objetos pertencem. Isso permite que o YOLO identifique objetos de diferentes classes em uma única imagem, enquanto também fornece informações sobre a localização precisa de cada objeto como mostrado na figura 7 (ALVES, 2020).

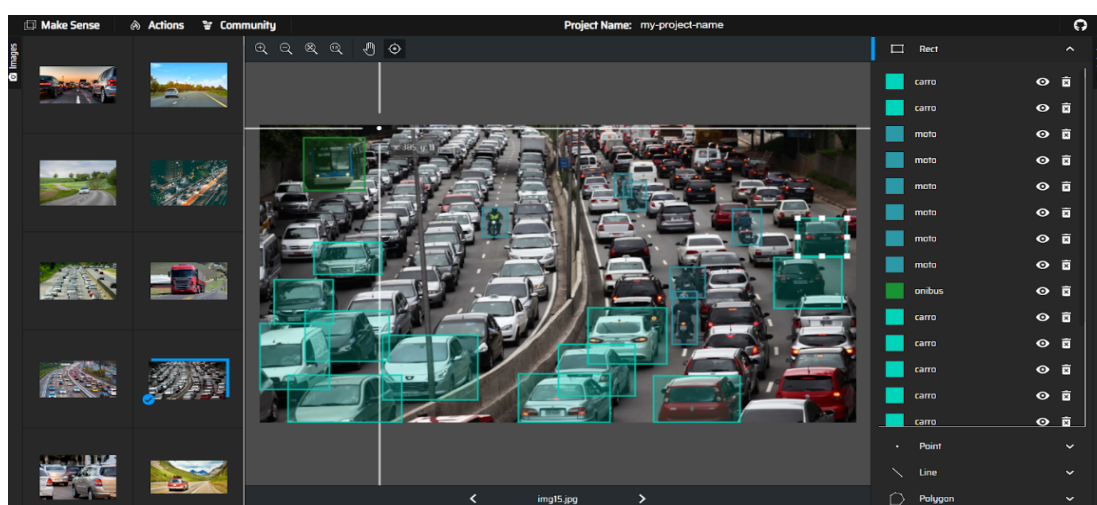
Figura 7 - Método de funcionamento do YOLO



Fonte: IA Expert Academy, 2020.

Para haver essa previsão é necessário fazer um treinamento para criar as classes desejadas e saber diferenciar os objetos presentes nas imagens. Para realizar o treinamento do sistema é necessário realizar uma coleta de dados a partir de banco de dados de interesse, entretanto, apenas as imagens no formato jpeg não é o suficiente, pois o YOLO necessita de um arquivo de mapeamento da área de interesse e para gerar esses arquivos temos duas opções sendo um manual buscando várias imagens de interesse e desenhando e classificando onde estão os objetos desejáveis utilizado o site *makesense.ia* como mostrado na figura 8.

Figura 8 - Processo de criação do dataset no makesense.



Fonte: Autores, 2023

A segunda opção o *Open Images Dataset v6* é considerada automática e mais rápida, pois utilizando uma aplicação por linha de comandos Python se escolhe quais classes de objetos devem ser pesquisados e o programa vai buscar no banco de dados já criado com mais de 600 classes catalogadas retomado tanto as imagens quanto o arquivo de mapeamento de objeto (figura 9), todavia o formato do mapeamento de objeto que é obtido é diferente do usado pelo YOLO então é preciso converter para o para uma estrutura a qual o YOLO possa ler e para isso é utilizado um código em Python que executa alguns cálculos. A conversão dos valores pode ser feita da forma como mostrado nas equações de 1 a 4:

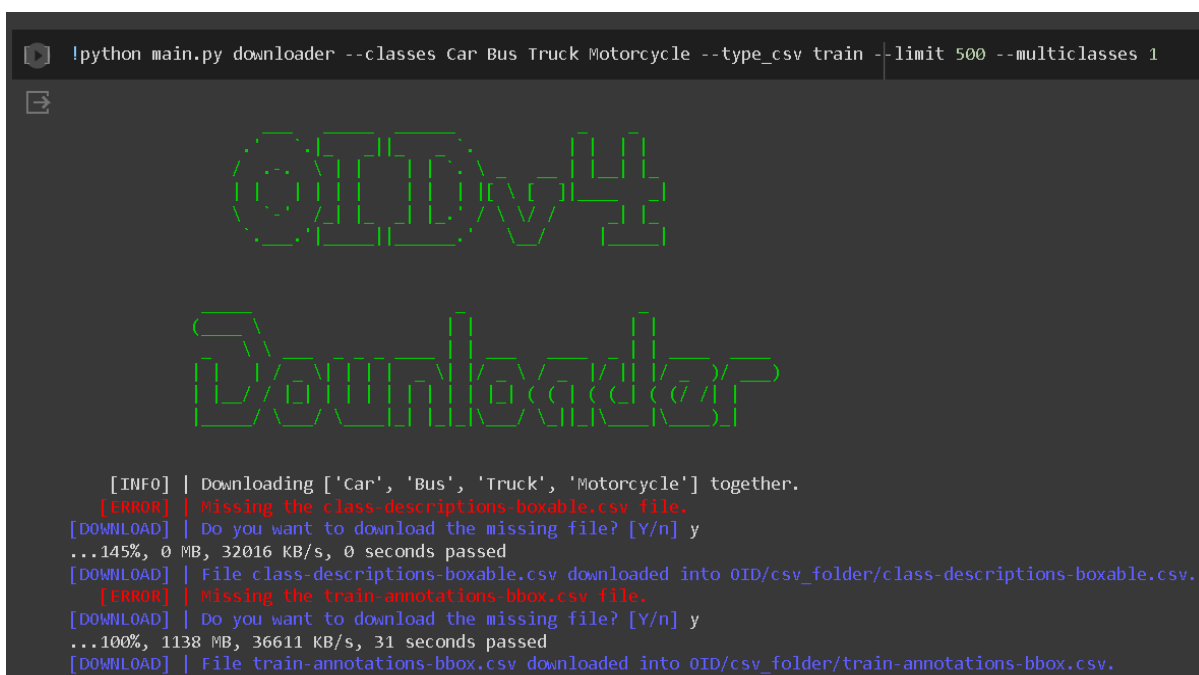
$$x = (((x \text{ fim} - x \text{ início})/2) + x \text{ início})/\text{largura imagem} \quad (1)$$

$$y = (((y \text{ fim} - y \text{ início})/2) + y \text{ início})/\text{altura imagem} \quad (2)$$

$$\text{largura} = (x \text{ fim} - x \text{ início})/\text{largura imagem} \quad (3)$$

$$\text{altura} = (y \text{ fim} - y \text{ início})/\text{altura imagem} \quad (4)$$

Figura 9 - Aquisição das imagens.



```
!python main.py downloader --classes Car Bus Truck Motorcycle --type_csv train --limit 500 --multiclass 1

[INFO] | Downloading ['Car', 'Bus', 'Truck', 'Motorcycle'] together.
[ERROR] | Missing the class-descriptions-boxable.csv file.
[DOWNLOAD] | Do you want to download the missing file? [Y/n] y
...145%, 0 MB, 32016 KB/s, 0 seconds passed
[DOWNLOAD] | File class-descriptions-boxable.csv downloaded into 0ID/csv_folder/class-descriptions-boxable.csv.
[ERROR] | Missing the train-annotations-bbox.csv file.
[DOWNLOAD] | Do you want to download the missing file? [Y/n] y
...100%, 1138 MB, 36611 KB/s, 31 seconds passed
[DOWNLOAD] | File train-annotations-bbox.csv downloaded into 0ID/csv_folder/train-annotations-bbox.csv.
```

Fonte: Autores, 2023

2.2.4.4 Darknet

A Darknet, que é uma estrutura de CNN popularmente usada para tarefas de visão computacional, como detecção de objetos, classificação de imagens e segmentação de imagens. Foi desenvolvida por Joseph Redmon e é conhecida por ser eficiente e altamente

personalizável. Há diversos componentes que integram a ferramenta, mais os principais componentes usados e como eles funcionam são:

Camadas Convolucionais: A base da Darknet é composta por várias camadas convolucionais. Essas camadas são responsáveis por aprender características abstratas das imagens de entrada. Elas aplicam filtros convolucionais para extrair informações relevantes, como bordas, texturas e formas, em várias escalas.

Camadas de Pooling: Após as camadas convolucionais, frequentemente se aplicam camadas de *pooling*, geralmente de *max-pooling*. Isso ajuda a reduzir a dimensionalidade da saída das camadas convolucionais e a preservar características essenciais da imagem.

Camadas Fully Connected (Densas): Após passar por várias camadas convolucionais e de *pooling*, as características extraídas são achatadas e alimentadas em camadas totalmente conectadas. Estas camadas são responsáveis por fazer previsões com base nas características extraídas. Em tarefas de classificação, por exemplo, as camadas densas podem ter um neurônio para cada classe possível.

Camadas de Ativação: Entre as camadas, funções de ativação, como ReLU, são comumente usadas para introduzir não linearidade no modelo, tornando-o capaz de aprender relacionamentos mais complexos nos dados.

Deteção de Objetos: A Darknet também é conhecida por seu uso em deteção de objetos. Para isso, são adicionadas camadas especiais, como as camadas YOLO. O YOLO é um algoritmo de deteção de objetos que divide a imagem de entrada em uma grade e faz previsões de caixas delimitadoras e classes para cada célula da grade. Isso permite a deteção eficiente de vários objetos em uma única passagem pela rede (REDMOND, 2016).

Função de Perda: Para treinar a rede, é usada uma função de perda que compara as previsões do modelo com as etiquetas (rótulos) verdadeiras dos dados de treinamento. A função de perda é ajustada para a tarefa específica, como deteção de objetos ou classificação de imagens.

Treinamento: Durante o treinamento, os pesos da rede são atualizados usando algoritmos de otimização, como o gradiente descendente, para minimizar a função de perda. Isso permite que a rede aprenda a realizar a tarefa desejada de forma eficaz.

Personalização: Uma das vantagens da Darknet é sua capacidade de personalização. Os usuários podem modificar a arquitetura, adicionar ou remover camadas, ajustar hiperparâmetros e adaptá-la para atender às necessidades específicas do projeto.

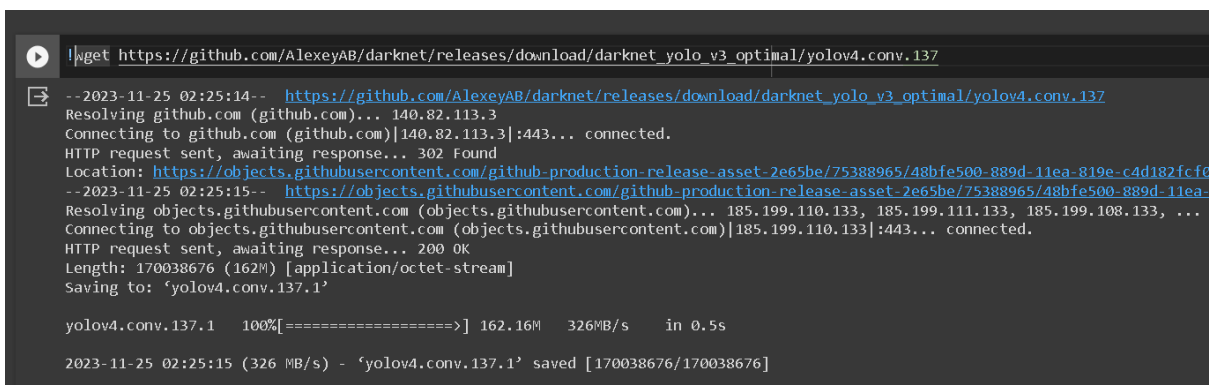
Em resumo, a arquitetura Darknet é uma estrutura flexível de rede neural convolucional usada para tarefas de visão computacional, como deteção de objetos e

classificação de imagens. Ela se destaca por sua eficiência e personalização, tornando-a uma escolha popular em projetos de visão computacional (REDMON, 2016).

2.2.4.4.1 Treinamento

Após a coleta de dados e a criação do dataset como foi descrito no item 2.2.2.3 é realizado o treinamento da rede neural e para tornar o processo mais rápido foi utilizado um modelo pré-treinado disponibilizado pelos próprios desenvolvedores do YOLO partindo o treinamento a partir da camada convolucional 137, pode ser baixada no github (figura 10).

Figura 10 - Download do modelo pré-treinado



```

!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137
--2023-11-25 02:25:14-- https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/75388965/48bfe500-889d-11ea-819e-c4d182fcf0
--2023-11-25 02:25:15-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/75388965/48bfe500-889d-11ea-8
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 170038676 (162M) [application/octet-stream]
Saving to: 'yolov4.conv.137.1'

yolov4.conv.137.1 100%[=====] 162.16M 326MB/s in 0.5s

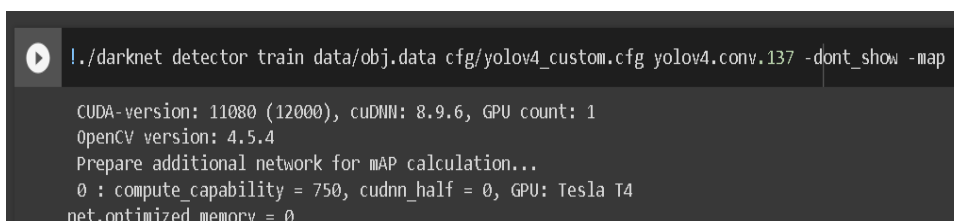
2023-11-25 02:25:15 (326 MB/s) - 'yolov4.conv.137.1' saved [170038676/170038676]

```

Fonte: Autores, 2023

A partir de então é realizado o treinamento em continuação ao modelo baixado, executando o comando para iniciar o treinamento da rede neural como pode ser visto na figura 11.

Figura 11 - Execução do treinamento da rede neural



```

!./darknet detector train data/obj.data cfg/yolov4_custom.cfg yolov4.conv.137 -dont_show -map
CUDA-version: 11080 (12000), cuDNN: 8.9.6, GPU count: 1
OpenCV version: 4.5.4
Prepare additional network for mAP calculation...
0 : compute_capability = 750, cudnn_half = 0, GPU: Tesla T4
net.optimized_memory = 0

```

Fonte: Autores, 2023

2.2.5 Comercialização do Produto

Para este projeto foi escolhido a venda do serviço, que proporciona uma maior flexibilidade e um relacionamento mais próximo com os possíveis compradores, permitindo que haja maior facilidade de adaptação e inovação, possibilidades de ajustes de acordo com a

demanda do cliente, além da possibilidade da escalabilidade que permite o crescimento sem haver necessidade de investimentos significativos em infraestrutura adicional.

2.2.6 Produtos Correlatos

Existem no mercado brasileiro alguns exemplos de produtos já existentes como o Semáforo Inteligente da Digicon, que opera de maneira semelhante. Nesse sistema, uma câmera é posicionada em um cruzamento, e a onda verde é liberada com base na movimentação de veículos apenas no sinal daquele cruzamento. Além disso, também temos o Tráfego da SERTTEL que utiliza sensores espalhados pela cidade para coletar dados e tomar decisões que melhorem o fluxo de trânsito.

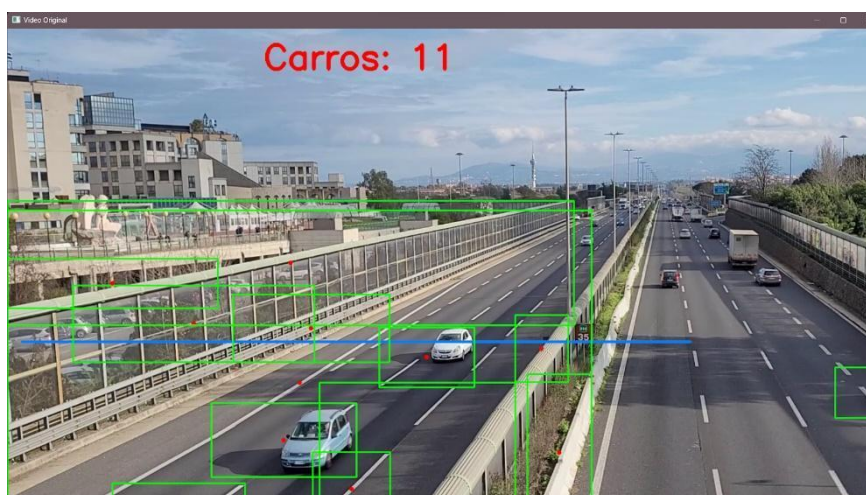
O SiTI entra com a proposta de aprimorar ideias como essas. Ele não apenas monitorará o trânsito, como nos exemplos mencionados, mas também gerenciará a organização do fluxo de veículos de forma autônoma em tempo real em uma determinada região. Isso será feito por meio de IA e ferramentas modernas baseadas em Python. O resultado será um sistema autônomo capaz de se adaptar a eventualidades e que também abrirá portas para futuras melhorias e integração de ferramentas adicionais, e assim adicionando uma contribuição significativa para o avanço do campo, promovendo uma gestão mais eficiente e segura do tráfego urbano.

3 RESULTADOS E DISCUSSÃO

O protótipo desenvolvido possui ainda algumas limitações, por ser uma tecnologia com uma certa complexidade, necessitando de mais conhecimentos para uma aplicação com um alto nível de confiança, onde poderá ser resolvido em uma continuação futura do trabalho.

Foram realizados alguns testes com vídeos baixados da internet e usando as tecnologias escolhidas citadas no item 2.2.2, onde inicialmente os testes ocorreram apenas com o OpenCV, pois não possui uma lógica muito rebuscada para programas. Contudo, com o decorrer da simulação realizada, foram notados problemas de inconsistência de informação durante o processo de execução do software no período de teste. É possível notar que na etapa de detecção há objetos indesejáveis sendo detectados como é possível de se identificar na figura 12, sendo possível observar a presença de *bounding boxes* em objetos diferentes dos quais se desejava.

Figura 12 - Programa apenas com OpenCV



Fonte: Autores, 2023

Isso se deve pelo fato que foram aplicados uma sequência de filtros (figura 13) nos métodos empregados, pois conseguimos obter informações a partir de bits (0 e 1).

Figura 13 - Filtros de cor

```

while True:
    ret, frame1 = cap.read() # Pega cada frame do video
    tempo = float(1 / delay)
    sleep(tempo) # Dá um delay entre cada processamento
    grey = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY) # Pega o frame e transforma para preto e branco
    blur = cv2.GaussianBlur(grey, (3, 3), 5) # Faz um blur para tentar remover as imperfeições da imagem
    img_sub = subtracao.apply(blur) # Faz a subtração da imagem aplicada no blur
    dilat = cv2.dilate(img_sub, np.ones((5, 5))) # "Engrossa" o que sobrou da subtração
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (
        5, 5)) # Cria uma matriz 5x5, em que o formato da matriz entre 0 e 1 forma uma elipse dentro
    dilatada = cv2.morphologyEx(dilat, cv2.MORPH_CLOSE, kernel) # Tenta preencher todos os "buracos" da imagem
    dilatada = cv2.morphologyEx(dilatada, cv2.MORPH_CLOSE, kernel)

```

Fonte: Autores, 2023

Vídeos coloridos RGB possuem três variáveis e cada uma delas tem um range de 0 a 255 e esses três valores juntos correspondem a uma cor figura 14.

Figura 14 - Cores em RGB

	R	G	B
Vermelho	255	0	0
Verde	0	255	0
Azul	0	0	255
Branco	255	255	255
Amarelo	255	255	0
Magenta	255	0	255
Ciano	0	255	255

Fonte: Eletrogate, 2021

Para podermos extrair as informações, precisamos aplicar um filtro de escala de cinza, nos permitindo obter apenas uma variável que varia de 0 a 255, em outras palavras ela varia do branco ou preto, entretanto para melhorar a informação é importante aplicar um filtro de desfoque para remover imperfeições e engrossamos as bordas para remover qualquer degradê tendo assim preto ou branco resultando no que pode ser visto na figura 15.

Figura 15 - Programa rodando a escala de cinza para detecção de objetos.



Fonte: Autores, 2023

Como pode ser observado na figura acima, há diversos problemas devido à movimentação da luz, sendo possível captar de maneira indesejável as sombras projetadas pelos veículos e os reflexos da luz solar nos prédios ao fundo.

Tendo em vista os problemas apresentados usando apenas o OpenCV seguimos para a implementação do YOLO com o OpenCV para resolver o problema do primeiro teste. Foi iniciada a criação do dataset de imagens relacionadas para a definir as classes desejadas (figura 16). Esse processo foi realizado por uma ferramenta do *Open Images Dataset* a qual facilitou esta criação.

Figura 16 - Downloader do dataset.

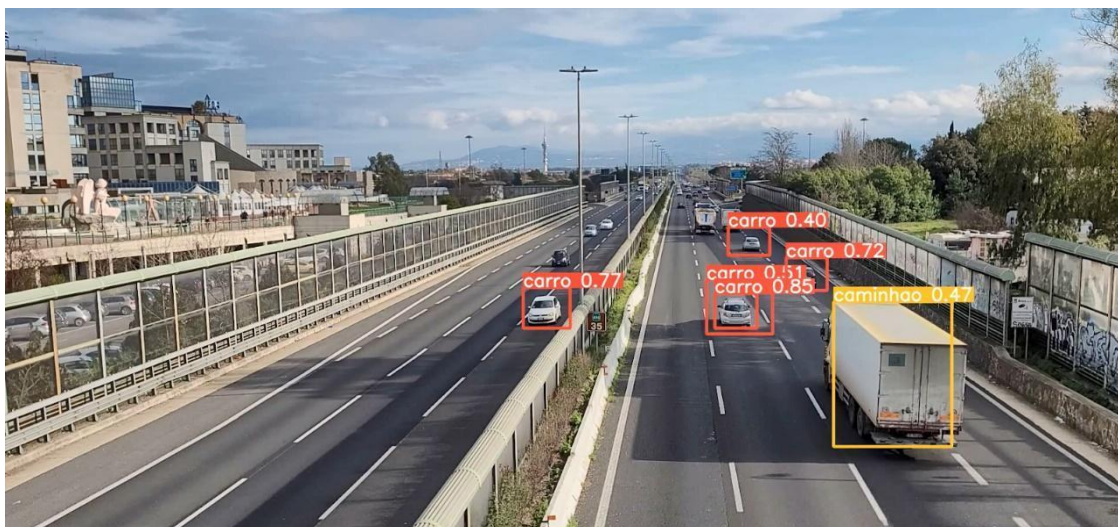
```
!python main.py downloader --classes Car Bus Truck Motorcycle --type_csv train --limit 500 --multiclass 1
```

```
!python main.py downloader --classes Car Bus Truck Motorcycle --type_csv test --limit 100 --multiclass 1
```

Fonte: Autores, 2023

No primeiro teste a partir da versão YOLO já foi possível notar uma resposta bem mais satisfatória, pois a rede neural consegue identificar as classes criadas e retornar uma porcentagem de certeza da classe, entretanto neste teste foi possível notar a presença de um fenômeno denominado *Non Maximum Suppression* que ocorre, pois, durante o processo de detecção de objetos são identificadas várias *bounding boxes*, no entanto, grande parte desses *bounding boxes* ou não possuem um objeto, ou estão compartilhando a mesma área de um mesmo objeto ao qual causa o fato presente na figura 17.

Figura 17 - Detecção com YOLO sem parâmetros de NMS.



Fonte: Autores, 2023

Para não ocorrer isso é necessário incrementar no código as funções THRESHOLD e NMS. Na primeira função é escolhido qual o valor mínimo de confiança que deve ser considerado, os valores mais baixos serão desconsiderados, a segunda função fica responsável por aceitar apenas o *bounding boxes* com maior confiança para um mesmo objeto detectado, assim retornando apenas um único *bounding boxes* com a maior taxa de certeza obtida.

Os resultados obtidos a partir da aplicação das funções antes citadas trouxeram resultados satisfatórios, reduzindo os erros de detecção drasticamente, como pode ser visto na figura 18.

Figura 18 - Detecção com YOLO com funções de correção de erros

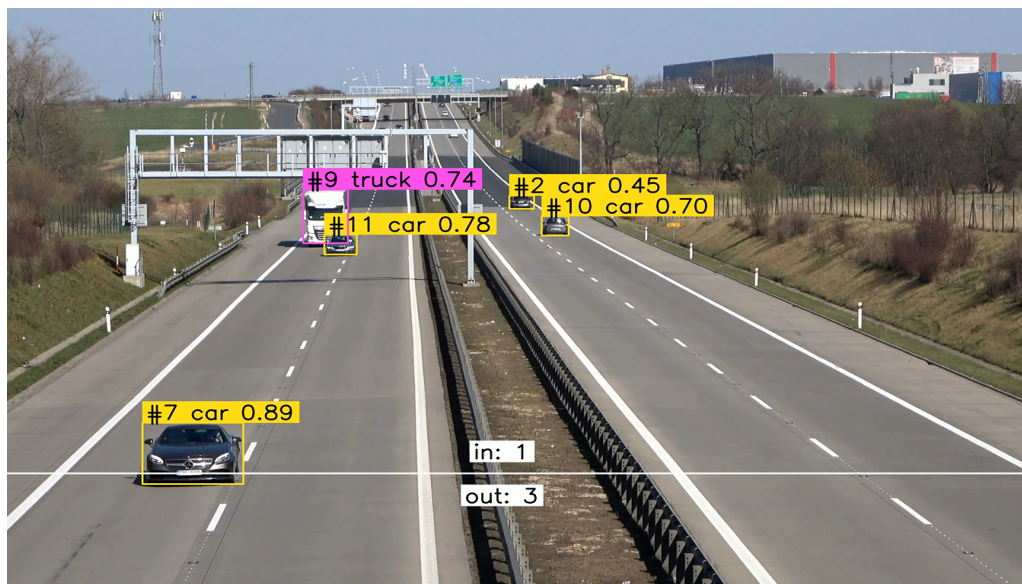


Fonte: Autores, 2023

A detecção de objetos possibilita até mesmo identificar veículos estacionados à esquerda fora da via, entretanto esses objetos para aplicação são errados, pois alteram a

contabilização dos veículos passando na estrada. Em um ambiente mais controlado onde não há objetos interferindo é obtido o que é esperado da contagem, como pode ser visto a figura 19.

Figura 19 - Detecção com YOLO com funções de correção de erros e contagem



Fonte: Autores, 2023

Nas imagens acima não há erros de detecção pelo fato de só haver a contagem dos veículos quando houver a passagem dos objetos pela linha de contagem.

O protótipo já proporciona a uma extração de dados dos veículos por meio da contagem e a identificação das classes, o que proporciona valores para serem aplicados em cálculos para o controle dos semáforos.

4 CONSIDERAÇÕES FINAIS

A presente pesquisa procurou investigar o impacto da implementação de sistemas de tráfego inteligente na mobilidade urbana, com um enfoque específico no contexto brasileiro. Ao examinar criticamente a evolução da mobilidade urbana no país e os desafios contemporâneos enfrentados por cidades em crescimento, ficou evidente que estratégias inovadoras são imperativas para lidar com os gargalos existentes.

A introdução de sistemas de tráfego inteligente surge como uma abordagem promissora para otimizar a circulação nas vias urbanas. Com a sinalização adaptativa, alimentada por algoritmos sofisticados e dados em tempo real, destaca-se como uma ferramenta crucial para lidar com as complexidades do tráfego nas cidades brasileiras. A discussão abordou temas como a eficiência aprimorada do transporte público, a redução do tempo de viagem e as possíveis implicações ambientais positivas decorrentes dos sistemas de tráfego inteligente.

As recomendações resultantes deste trabalho podem servir como um guia valioso para a implementação eficaz de sistemas de tráfego inteligente em cidades brasileiras, visto que o sistema consegue identificar e contabilizar os veículos com bastante precisão e tomar decisões através das informações coletadas, contribuindo para a construção de ambientes urbanos mais eficientes, sustentáveis e habitáveis. Os achados deste estudo têm implicações práticas significativas para gestores urbanos, formuladores de políticas públicas e profissionais envolvidos na melhoria da mobilidade urbana, não só mostrando os resultados eficientes como também demonstrando os problemas encontrados como tivemos com o OpenCV que demonstrou ser ineficiente na construção da IA, mas se mostrou eficiente junto a outras ferramentas como o YOLO para o desenvolvimento do projeto.

Em última análise, esta pesquisa oferece uma contribuição para a compreensão dos desafios e oportunidades associados à mobilidade urbana no Brasil, abrindo portas para investigações mais aprofundadas e a implementação futuras por melhorias como a identificação de infrações de trânsito e aplicação de multas automáticas identificadas pelo próprio sistema autônomo, soluções conscientes que visem aprimorar a qualidade de vida nas cidades brasileiras. O caminho para cidades mais inteligentes e eficientes está intrinsecamente ligado à capacidade de inovação, adaptação e colaboração em busca de soluções sustentáveis para os desafios da mobilidade urbana contemporânea.

5 REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Gabriel. Detecção de Objetos com YOLO–Uma abordagem moderna. **Expert Academy**. [SI], v. 13, 2020.

ALVES, José Eustáquio Diniz; GALIZA, Francisco. **Demografia e Economia: nos 200 anos da Independência do Brasil e cenários para o século XXI**. 1ª edição. Editora Escola de Negócios e Seguros, 2022.

ARAÚJO, Marley Rosana Melo de et al. Transporte público coletivo: discutindo acessibilidade, mobilidade e qualidade de vida. **Psicologia & Sociedade**, v. 23, p. 574-582, 2011.

BARBOSA, Marina. **População dos grandes centros perde em média 21 dias do ano no trânsito, aponta CNDL / SPC Brasil**. CNLD. Disponível em: <https://site.cndl.org.br/populacao-dos-grandes-centros-perde-em-media-21-dias-do-ano-no-trnsito-aponta-cndl-spc-brasil/>. Acesso em: 15 set. 2023.

BEZERRA, Juliana. **Revolução Industrial: o que foi (resumo)**. Toda Matéria, [s.d.]. Disponível em: <https://www.todamateria.com.br/revolucao-industrial/>. Acesso em: 12 nov. 2023

BRADSKI, G.; KAEHLER, A. (2008) **Learning OpenCV**. Sebastopol: O’Reilly Media Inc. 1 p. ISBN: 978-0-596-51613-0.

BRASIL. [Constituição (1988)]. **Constituição da República Federativa do Brasil**. Diário Oficial da União - Seção 1 - 04/01/2012, Página 1.

CNDL BRASIL. **População dos grandes centros perde em média 21 dias do ano no trânsito, aponta CNDL / SPC Brasil**. Disponível em <<https://site.cndl.org.br/populacao-dos-grandes-centros-perde-em-media-21-dias-do-ano-no-trnsito-aponta-cndl-spc-brasil/>>. Acesso em: 10 de setembro de 2023.

CORTESE, Tatiana Tucunduva Philippi et al. Tecnologias e sustentabilidade nas cidades. **Estudos Avançados**, v. 33, p. 137-150, 2019.

DATHEIN, Ricardo. Inovação e Revoluções Industriais: uma apresentação das mudanças tecnológicas determinantes nos séculos XVIII e XIX. **Publicações DECON Textos Didáticos**, v. 2, p. 2003, 2003.

FLEURY, Maria Tereza Leme; FLEURY, Afonso. **Construindo o conceito de competência**. 2009.

IBGE. **Estimativas de população enviadas ao TCU**. [2021]. 1 tabela. Disponível em: https://ftp.ibge.gov.br/Estimativas_de_Populacao/Estimativas_2021/POP2021_20230710.pdf. Acesso em: 15/11/2023.

JOSÉ, M. et al. **OCDE e o Brasil (Publicação Preliminar)**. Brasília: Ipea, 2023. 1052p.

LIMA, Tiago de. **Uma análise de soft skills na visão dos profissionais da engenharia de software**. 2018.

MAGALHÃES, Regina; VENDRAMINI, Annelise. **Os impactos da quarta revolução industrial**. 2018.

MARTINS. **Controlando fitas de LED RGB com o arduino**. Disponível em: <https://blog.eletrogate.com/controlando-fitas-de-led-rgb-com-o-arduino-parte-1/> Acesso em: 26 de outubro de 2023.

MARQUES, Rodrigues. **OpenCV: Uma breve introdução à visão computacional com python**. Disponível em: <https://www.cedrotech.com/blog/opencv-uma-breve-introducao-visao-computacional-com-python/#>. Acesso em: 16 de outubro de 2023.

MUMFORD, Lewis. A cidade na história: suas origens, desenvolvimento e perspectivas. **Tradução de Neil R. da Silva**, v. 2, 1982.

OBSERVATÓRIO DAS METRÓPOLES. **Mapa da motorização individual no Brasil 2019**. [2019]. 1 figura. Disponível em: https://www.observatoriodasmetroles.net.br/wp-content/uploads/2019/09/mapa_moto2019v2.pdf. Acesso em: 29/09/2023.

REDMON, **Darknet: Open Source Neural Networks in C**. Disponível em: <https://pjreddie.com/darknet/> Acesso em: 30 de outubro de 2023.

RODRIGUES, J. et al. **Mapa da Motorização Individual no Brasil 2019**. Rio de Janeiro: Observatório das Metrôpoles, 2019.

SANTANA, Bruno. **O Que é Python: Conheça Uma das Linguagens de Programação mais Populares do Mundo**. Disponível em: <https://www.hostinger.com.br/tutoriais/python-o-que-e>. Acesso em: 16 de outubro de 2023.

SCHNEIDER, Elmir. História do sistema rodoviário brasileiro. **Revista Escola DetranRS**, v. 1, n. 1, p. 45-58, 2021.

SOFIA, D. **Sinais de Trânsito Inteligentes**. Tese (Dissertação de Mestrado em Matemática e Computação) - Universidade do Minho, Escola de Ciências. Braga, p. 74, 2020.

SOUSA, Rafaela. **"Urbanização"; Brasil Escola**. Disponível em: <https://brasilescuela.uol.com.br/brasil/urbanizacao.htm>. Acesso em: 10 de setembro de 2023.