

**UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ INSTITUTO DE  
GEOCIÊNCIAS E ENGENHARIAS  
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO**

LUÍS GUILHERME GOIS MATOS

**PROPOSTA DE UM MODELO DE SISTEMA INTELIGENTE DE SEMÁFORO COM  
VISÃO COMPUTACIONAL**

MARABÁ-PA  
2025

LUIS GUILHERME GOIS MATOS

**PROPOSTA DE UM MODELO DE SISTEMA INTELIGENTE DE SEMÁFORO COM  
VISÃO COMPUTACIONAL**

Projeto Final de Curso, apresentado à Universidade Federal do Sul e Sudeste do Pará, como parte dos requisitos necessários para obtenção do Título de Bacharelado em Engenharia da Computação.

Orientador: Prof. Dr. José Carlos da Silva

Coorientador: Prof. Dr. Haroldo Gomes Barroso Filho

MARABÁ-PA

2025

LUIS GUILHERME GOIS MATOS

**PROPOSTA DE UM MODELO DE SISTEMA INTELIGENTE DE SEMÁFORO COM  
VISÃO COMPUTACIONAL**

Projeto Final de Curso

Projeto Final de Curso, apresentado à Universidade Federal do Sul e Sudeste do Pará, como parte dos requisitos necessários para obtenção do Título de Bacharelado em Engenharia da Computação.

**BANCA QUALIFICADORA:**

---

Prof. Dr. José Carlos da Silva – Orientador (Presidente)

---

Prof Dr. Haroldo Gomes Barroso Filho – Coorientador (Membro – UFMA)

---

Prof Dr. João Victor Costa Carmona (Membro – UNIFESSPA )

---

Prof Dr. Elton Rafael Alves (Membro – UNIFESSPA)

MARABÁ-PA

2025

## **AGRADECIMENTOS**

Primeiramente, expresso minha profunda gratidão à minha avó Deusdet, que foi uma segunda mãe para mim e sempre esteve ao meu lado, me dando forças em todos os momentos. Seu maior sonho era acompanhar minha trajetória acadêmica, e sei que, onde quer que esteja, ela estará orgulhosa desta conquista.

Aos meus pais, Donner e Marcela, agradeço pelo incentivo incondicional aos meus estudos e, especialmente, à minha mãe, por sempre ter feito de tudo por mim e por me moldar para ser a pessoa que sou hoje.

Ao meu irmão, Victor Gois Moreira, sou imensamente grato por sua contribuição ao meu projeto de conclusão de curso e por sempre estar ao meu lado, me apoiando em cada etapa dessa jornada.

Agradeço ao meu tio Denner, ao meu avô Aristeu e à minha tia Dani, por sempre me acolheram e cuidaram de mim com tanto carinho, sem eles, nada disso seria possível. Obrigado por tudo.

Aos meus amigos de curso, deixo minha gratidão, em especial ao Déric, que foi um grande amigo ao longo da minha trajetória acadêmica e pessoal. Também agradeço aos membros do grupo "Piscina do Luis", que tornaram essa jornada mais leve e inesquecível.

Sou extremamente grato ao meu orientador, Prof. Dr. José Carlos, que, além de um excelente professor, sempre foi um grande amigo dentro da universidade, me ajudando em tudo que estivesse ao seu alcance. Agradeço também ao Prof. Dr. Haroldo, por ter aceitado ser meu coorientador neste projeto, mesmo diante das dificuldades e barreiras impostas pela distância e sua atuação em outra universidade.

Por fim, mas não menos importante, agradeço à minha namorada, Clara Maria, por estar ao meu lado em todos os momentos, oferecendo seu apoio incondicional e amor inabalável. Eu te amo.

## RESUMO

As travessias de pedestres em semáforos no Brasil representam um desafio para a mobilidade urbana, pois os tempos fixos programados nos semáforos não levam em conta a demanda real de pedestres. Esse fator compromete a segurança, principalmente de pessoas com mobilidade reduzida, que muitas vezes não conseguem concluir a travessia a tempo. Para superar esse problema, este trabalho propõe uma simulação de um sistema semafórico inteligente baseado em visão computacional, capaz de ajustar dinamicamente o tempo do sinal de acordo com o número e o perfil dos pedestres aguardando para atravessar. Além disso, foi introduzido um aviso sonoro para auxiliar motoristas e pedestres na identificação do status do semáforo. A abordagem utiliza o YOLOv11, um algoritmo de detecção de objetos em tempo real, para identificar indivíduos na travessia de pedestres e detectar possíveis dificuldades de mobilidade. O sistema atribui pesos diferentes às classes detectadas, incluindo pessoas, muletas, carrinhos de bebê e cadeiras de rodas, ajustando de forma adaptativa o tempo semafórico para garantir uma travessia mais segura e eficiente. O projeto foi validado por meio de uma simulação desenvolvida em Python, utilizando um dataset personalizado para garantir precisão e diversidade na detecção. Embora os testes tenham sido satisfatórios e demonstrado a capacidade do sistema de adaptar dinamicamente a programação dos semáforos, a aplicabilidade em um ambiente real exige estudos adicionais para avaliar desafios técnicos, eficácia prática e impacto real na mobilidade urbana.

**Palavras-chave:** Semáforo inteligente, visão computacional, YOLO, mobilidade urbana, trânsito eficiente.

## ABSTRACT

Pedestrian crossings at traffic lights in Brazil pose a challenge for urban mobility, as fixed signal timing does not account for real pedestrian demand. This limitation compromises safety, especially for individuals with reduced mobility, who often fail to complete the crossing in time. To address this issue, this study proposes an intelligent traffic light control system based on computer vision, capable of dynamically adjusting the signal duration according to the number and profile of pedestrians waiting to cross. Additionally, an auditory alert has been introduced to assist both drivers and pedestrians in identifying the traffic light status. The approach utilizes YOLOv11, a real-time object detection algorithm, to identify individuals at pedestrian crossings and detect potential mobility impairments. The system assigns different weights to detected classes, including pedestrians, crutches, baby strollers, and wheelchairs, adjusting the traffic light timing adaptively to ensure a safer and more efficient crossing. The project was validated through a Python-based simulation, using a custom dataset to ensure detection accuracy and diversity. Although initial tests were satisfactory and demonstrated the system's ability to dynamically adjust traffic light programming, its application in a real-world environment requires further studies to assess technical challenges, practical effectiveness, and its actual impact on urban mobility.

**Keywords:** Intelligent traffic light, computer vision, YOLO, urban mobility, efficient traffic.

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>10</b>
1.1 Contexto.....	10
1.2 Objetivo Geral.....	11
1.3 Objetivos Específicos.....	11
1.4 Justificativa.....	12
1.5 Organização do Trabalho.....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
2.1 Inteligência Artificial e Aprendizado de Máquina.....	15
2.2 Visão Computacional.....	17
2.3 Redes Neurais Convolucionais (CNNs).....	18
2.3.1 Camada Convolucional.....	19
2.3.2 Camada de Pooling.....	20
2.4 Frameworks de Visão Computacional: YOLOv11.....	21
<b>3. TRABALHOS CORRELATOS.....</b>	<b>23</b>
3.1 Metodologia para Controle de Semáforos Utilizando Processamento de Imagens para Reconhecimento da Quantidade de Carros.....	23
3.2 Controle de Semáforos em Tempo Real: A Experiência de São Paulo.....	23
3.3 Análise de Tráfego Urbano: Visão Computacional e Inteligência Artificial para Otimização de Semáforos.....	24
3.4 Uma Revisão Abrangente das Arquiteturas YOLO em Visão Computacional.....	24
3.5 Análise da Acessibilidade dos Espaços Urbanos para Usuários com Mobilidade Reduzida.....	24
3.6 Análise da Acessibilidade dos Espaços Urbanos para Usuários com Mobilidade Reduzida.....	25
3.7 Correlação dos trabalhos.....	25
<b>4. METODOLOGIA.....</b>	<b>28</b>
4.1 Sistema.....	28
4.2 Pesquisa de Campo.....	29
4.2.1 Anotação das Imagens (Rotularização das imagens).....	29
4.2.2 Unificação do Dataset.....	30
4.3 Desenvolvimento da Simulação do Sistema de Semáforo Inteligente.....	30
4.4 Pesos e Parâmetros da simulação.....	31
4.7 Modularização do Código.....	32
4.8 Sinal Sonoro e Acessibilidade.....	32
<b>5. RESULTADOS E DISCUSSÃO.....</b>	<b>33</b>
5.1 Pesquisa de Campo.....	34
5.1.1 Construção do Dataset.....	35
5.1.2 Treinamento do Modelo YOLOv11m - Métricas.....	37
5.2 Testes.....	44
5.3 Simulação do Sistema.....	46
<b>6. CONCLUSÃO.....</b>	<b>48</b>
6.1 Limitações e Trabalhos Futuros.....	48
6.2 Conclusão final.....	48

**APÊNDICES**

<b>APÊNDICE A – QUESTIONÁRIO ESTRUTURADO.....</b>	<b>50</b>
<b>APÊNDICE B – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO.....</b>	<b>51</b>
<b>APÊNDICE C – LÓGICA DE FUNCIONAMENTO DA SIMULAÇÃO DO SEMÁFORO.....</b>	<b>52</b>
<b>REFERÊNCIAS.....</b>	<b>54</b>

## LISTA DE FIGURAS

Figura 1 - Representação de um sistema que utiliza visão computacional para detecção de diferentes classes no tráfego urbano.....	16
Figura 2 - Exemplo de Fluxograma de Processamento de Machine Learning.....	18
Figura 3 - Arquitetura de Rede Neural para Classificação de Imagens (CNN).....	19
Figura 4 - Diagrama de Fluxo de uma CNN para Processamento de Imagens.....	20
Figura 5 - Operação de pooling em uma Rede Neural Convolutiva (CNN).....	20
Figura 6 - gráfico de desempenho comparativo entre diferentes versões do modelo YOLO.	22
Figura 7 - Fluxo do sistema do semáforo inteligente.....	26
Figura 8 - Estrutura das Labels após o tratamento dos dados.....	28
Figura 09 - Código demonstrando a utilização da biblioteca playsound.....	29
Figura 10 - Gráfico da pesquisa.....	31
Figura 11 - Total de 19 pessoas identificadas utilizando o cocodataset.....	32
Figura 12 - Total de 51 pessoas identificadas utilizando o dataset customizado.....	35
Figura 13 - Código demonstrando lógica das condições para o funcionamento correto do semáforo.....	37
Figura 14 - Demonstração do sinal verde.....	38
Figura 15 - Demonstração do sinal vermelho.....	38
Figura 16 - Plot do gráfico F1 gerado ao finalizar treinamento do dataset, usado para avaliar o equilíbrio entre precisão e recall.....	39
Figura 17 - Plot da matriz de confusão normalizada, mostrando a precisão de detecção por classe.....	40
Figura 18 - Demonstração da identificação de muletas utilizando o dataset customizado....	40
Figura 19 - Demonstração da identificação de carrinhos de bebê utilizando o dataset customizado.....	41
Figura 20 - Estado do semáforo contabilizando uma pessoa para a travessia.....	41
Figura 21 - Estado do semáforo amarelo para representar atenção.....	41
Figura 22 - Estado do semáforo fechado, possibilitando a travessia dos pedestres.....	42
Figura 23 - Demonstração do sinal vermelho.....	42
Figura 24 - Plot da matriz de confusão normalizada.....	43
Figura 25 - Plot da matriz de confusão normalizada.....	43
Figura 26 - Estado do semáforo contabilizando uma pessoa para a travessia.....	46
Figura 27 - Estado do semáforo amarelo para representar atenção.....	46
Figura 28 - Estado do semáforo fechado, possibilitando a travessia dos pedestres.....	47
Figura 29 - Simulação do sistema de semáforo inteligente.....	50
Figura 30 - Teste da visão computacional na detecção de pedestres.....	51
Figura 31 - Código demonstrando lógica das condições para funcionamento correto do semáforo.....	53

# 1 INTRODUÇÃO

## 1.1 Contexto

No Brasil, especialmente nas grandes capitais, a forma como os semáforos são programados e as vias são estruturadas ainda prioriza o fluxo de veículos, muitas vezes deixando de lado as necessidades dos pedestres. A mobilidade urbana, no entanto, deve ser pensada para todos, independentemente de sua posição no trânsito ou condição física. Quando essa adaptação não acontece, as consequências são claras: acessibilidade comprometida, insegurança na travessia e tempos de espera excessivos, o que pode tornar a simples ação de atravessar a rua um risco (Silva et al., 2023).

Um estudo do Instituto Corrida Amiga revelou que, na cidade de São Paulo, o tempo destinado à travessia de pedestres nem sempre é suficiente. Em algumas áreas, a espera ultrapassa quatro minutos, enquanto o tempo para cruzar a via é de apenas seis segundos (OBSERVATÓRIO DO TERCEIRO SETOR, 2021). Esses números preocupam, sobretudo quando consideramos que, de acordo com a Organização Mundial da Saúde (OMS), os pedestres representam 21% das mortes globais no trânsito, o que reforça a urgência de medidas que tornem as travessias mais seguras (ORGANIZAÇÃO MUNDIAL DA SAÚDE, 2023).

Além disso, a falta de adaptação dos tempos semaforicos impacta de maneira significativa pessoas com deficiência, como usuários de cadeiras de rodas e muletas. Segundo a Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD) do Instituto Brasileiro de Geografia e Estatística (IBGE), o Brasil possui aproximadamente 18,6 milhões de pessoas com deficiência, representando 8,9% da população nacional. Esse grupo enfrenta desafios adicionais de acessibilidade, incluindo maior tempo necessário para atravessar ruas e dificuldades impostas por infraestruturas urbanas pouco inclusivas (IBGE; MDHC, 2023). A inadequação dos tempos semaforicos pode fazer com que esses pedestres fiquem presos no meio da travessia, aumentando o risco de acidentes e dificultando sua mobilidade no espaço urbano.

Apesar de algumas iniciativas municipais para melhorar a segurança viária, como o Pedestre Seguro, implantado pela Companhia de Engenharia de Tráfego (CET), o problema persiste. A situação é ainda mais grave para pessoas com mobilidade reduzida, que muitas vezes não conseguem concluir a travessia a tempo.

Além disso, um cenário comum é o de pedestres que ficam presos em ilhas centrais, quando o tempo semafórico não é suficiente para que atravesse completamente a via, expondo-os ao tráfego de veículos (Santos; Melo; Ferreira, 2020).

Algumas cidades vêm apostando em soluções tecnológicas para otimizar a mobilidade urbana. Barra do Garças (MT), por exemplo, implementou semáforos inteligentes com botões para travessia de pedestres. Nesse modelo, o semáforo permanece verde para os veículos e só fecha quando um pedestre pressiona o botão para atravessar (BARRA DO GARÇAS, 2023). Embora esse recurso represente um avanço, ele ainda prioriza o fluxo de veículos e depende da ação do pedestre. Isso pode ser um problema para pessoas com dificuldades motoras ou, em um cenário pós-pandemia, levantar preocupações quanto ao contato físico com superfícies em locais de grande circulação (Carvalho; Souza, 2021).

Diante desse cenário, este estudo propõe uma simulação de sistema inteligente de travessia, que não apenas ajusta dinamicamente o tempo semafórico de acordo com a demanda real dos pedestres, mas também elimina a necessidade de qualquer ação manual por parte dos pedestres. Baseado em visão computacional, o sistema consegue identificar a quantidade e o perfil de alguns pedestres aguardando a travessia, adaptando o tempo do semáforo para garantir mais acessibilidade e segurança viária.

## **1.2 Objetivo Geral**

O objetivo principal foi simular um sistema de semáforo inteligente baseado em visão computacional e aprendizado profundo, capaz de ajustar dinamicamente o tempo semafórico de acordo com a quantidade e o perfil dos pedestres.

## **1.3 Objetivos Específicos**

**1.3.1** Implementar o modelo YOLOv11 com um dataset customizado para a detecção de pedestres e suas respectivas condições de mobilidade (utilização de muletas, transporte de carrinho de bebê e cadeiras de rodas), garantindo um reconhecimento eficiente e em tempo real.

**1.3.2** Testar o modelo em diferentes tipos de mídia, incluindo fotos, vídeos gravados, transmissões ao vivo (lives) e câmeras em tempo real (webcam), avaliando sua robustez em condições adversas.

**1.3.3** Realizar pesquisa de campo para identificar os desafios reais enfrentados pelos pedestres, coletando informações que possam contribuir para a calibração e aprimoramento do sistema.

**1.3.4** Desenvolver uma simulação visual do semáforo, permitindo uma demonstração interativa do funcionamento do sistema e da adaptação dinâmica dos tempos semafóricos.

**1.3.5** Criar e implementar um cálculo de pesos para ajustar o tempo de travessia do semáforo com base na quantidade e no perfil dos pedestres detectados, garantindo maior segurança e acessibilidade.

**1.3.6** Desenvolver um sinal sonoro de aviso para indicar a mudança do estado do semáforo (abertura e fechamento), garantindo maior acessibilidade para pessoas com deficiência visual ou com dificuldades de percepção visual do semáforo.

#### **1.4 Justificativa**

A segurança dos pedestres em travessias urbanas é uma preocupação crescente na mobilidade urbana brasileira. Os acidentes de trânsito continuam sendo uma das principais causas de morte no país, e a falta de medidas eficazes para proteção dos pedestres aumenta o risco de atropelamentos.

Embora existam iniciativas de visão computacional aplicadas ao trânsito, como detecção de sonolência ao volante e gestão de tráfego veicular, ainda há uma lacuna significativa no desenvolvimento de tecnologias focadas exclusivamente na segurança dos pedestres. A utilização de inteligência artificial na mobilidade urbana já vem demonstrando resultados promissores, com aplicações voltadas para previsão de congestionamentos e ajuste automático de tempos semafóricos, auxiliando na redução de acidentes e otimização do fluxo viário (REVISTA FÓRUM TRANSPORTE, 2023).

Diante desse cenário, este projeto pode permitir a segurança nas travessias de pedestres, ao mesmo tempo em que melhora a fluidez do tráfego urbano. Para isso, propõe-se utilizar visão computacional e uma simulação semafórica, para que o tempo semafórico seja ajustado dinamicamente de acordo com a demanda de pedestres identificadas na simulação. Essa abordagem visa demonstrar uma proposta que, se implementada no mundo real, tem potencial para contribuir na redução do número de atropelamentos e na promoção de um trânsito mais acessível e eficiente, aumentando a segurança tanto para pedestres quanto para motoristas. Entretanto, estudos futuros e testes em ambiente real seriam necessários para validar esses impactos de forma concreta.

### **1.5 Organização do Trabalho**

Este trabalho está estruturado em cinco capítulos, organizados para que a compreensão do estudo ocorra de forma clara e progressiva. No primeiro capítulo, apresenta-se a introdução, onde são expostos o contexto, a motivação e os objetivos do estudo. Também são abordadas as principais dificuldades enfrentadas por pedestres ao atravessar faixas semaforizadas, especialmente aqueles com mobilidade reduzida, e a necessidade de soluções tecnológicas para tornar o trânsito mais seguro e acessível.

O segundo capítulo é dedicado à fundamentação teórica, reunindo conceitos essenciais para compreender os pilares deste estudo. São explorados temas como visão computacional, inteligência artificial aplicada ao trânsito e redes neurais convolucionais, além de uma revisão de trabalhos anteriores que já aplicaram essas tecnologias ao tráfego urbano. Essa análise ajuda a evidenciar os avanços na área e as lacunas existentes, justificando a necessidade da abordagem proposta neste projeto.

No terceiro capítulo, trabalhos correlatos, onde se é explorado e comparado trabalhos de outras pessoas que têm um grau de proximidade com o trabalho proposto.

No quarto capítulo, a metodologia adotada no desenvolvimento do projeto é descrita detalhadamente. Cada etapa do processo é explicada, incluindo:

- Pesquisa de campo, realizada para compreender as dificuldades reais enfrentadas pelos pedestres em travessias semaforizadas. Esse levantamento foi

essencial para direcionar ajustes no modelo e garantir que ele atenda às demandas reais da mobilidade urbana.

- Construção do dataset no Roboflow, garantindo que as imagens fossem organizadas, rotuladas corretamente e livres de conflitos nos índices.
- Unificação dos dados em um único dataset, seguido da configuração de um novo arquivo data.yaml, ajustado para o treinamento da rede neural.
- Treinamento do modelo, resultando na geração da pasta runs com o arquivo best.pt (Yolov11m customizado com o novo dataset), que contém os melhores pesos obtidos no aprendizado da IA.
- Testes com diferentes tipos de mídia, como imagens, vídeos gravados, transmissões ao vivo (lives) e identificação em tempo real através de câmeras (webcam acoplada ao computador), para avaliar a precisão da detecção. Durante essa fase, foi observado um desempenho abaixo do esperado na classe "cadeira de rodas", o motivo será debatido mais a frente.

O quarto capítulo apresenta a análise e discussão dos resultados, onde são expostos os dados coletados durante os testes e as decisões tomadas para melhorar o desempenho do modelo. Além disso, este capítulo descreve a implementação da simulação do semáforo, que inclui:

- Interface visual utilizando OpenCV (cv2) para representar o funcionamento do semáforo.
- Lógica de transição dos sinais, simulando as mudanças entre verde, amarelo e vermelho.
- Cálculo de pesos para ajustar o tempo de travessia, considerando a quantidade e o perfil dos pedestres detectados.
- Implementação de um aviso sonoro, tornando a travessia mais acessível para pessoas com deficiência visual ou dificuldades de percepção do semáforo.

O quinto capítulo foi construído para discussões referentes aos resultados obtidos no projeto.

Por fim, o sexto capítulo reúne as considerações finais, apresentando um balanço do projeto, suas contribuições e suas limitações. Também são discutidas propostas para trabalhos futuros, como a necessidade de expandir o dataset para melhorar a detecção de classes menos representadas e a possibilidade de implementação do sistema em cenários reais.

## **2 FUNDAMENTAÇÃO TEÓRICA**

A fundamentação teórica deste trabalho aborda os conceitos essenciais para o desenvolvimento do sistema inteligente semafórico baseado em visão computacional. Primeiramente, são discutidos os desafios da mobilidade urbana e segurança viária, enfatizando a necessidade de soluções inovadoras para travessias de pedestres, considerando dados estatísticos sobre acidentes de trânsito e o impacto da acessibilidade inadequada.

Em seguida, são explorados os sistemas semafóricos inteligentes, destacando tecnologias que permitem a adaptação dinâmica do tempo de travessia com base na demanda real. Também são abordadas aplicações da visão computacional no trânsito, evidenciando o papel das redes neurais convolucionais (CNNs) na detecção de objetos, com ênfase no modelo YOLOv11 utilizado no projeto.

Além disso, o capítulo detalha a construção e o treinamento do dataset, explicando o uso do Roboflow para organização das imagens, os ajustes necessários na rotulagem e o processo de unificação do conjunto de dados. Por fim, são descritas a implementação da simulação do sistema semafórico e a integração de cálculos de pesos para ajustes no tempo semafórico, garantindo maior segurança e acessibilidade na travessia e estabelecendo a base teórica necessária para compreender as decisões técnicas adotadas ao longo do desenvolvimento do projeto.

### **2.1 Inteligência Artificial e Aprendizado de Máquina**

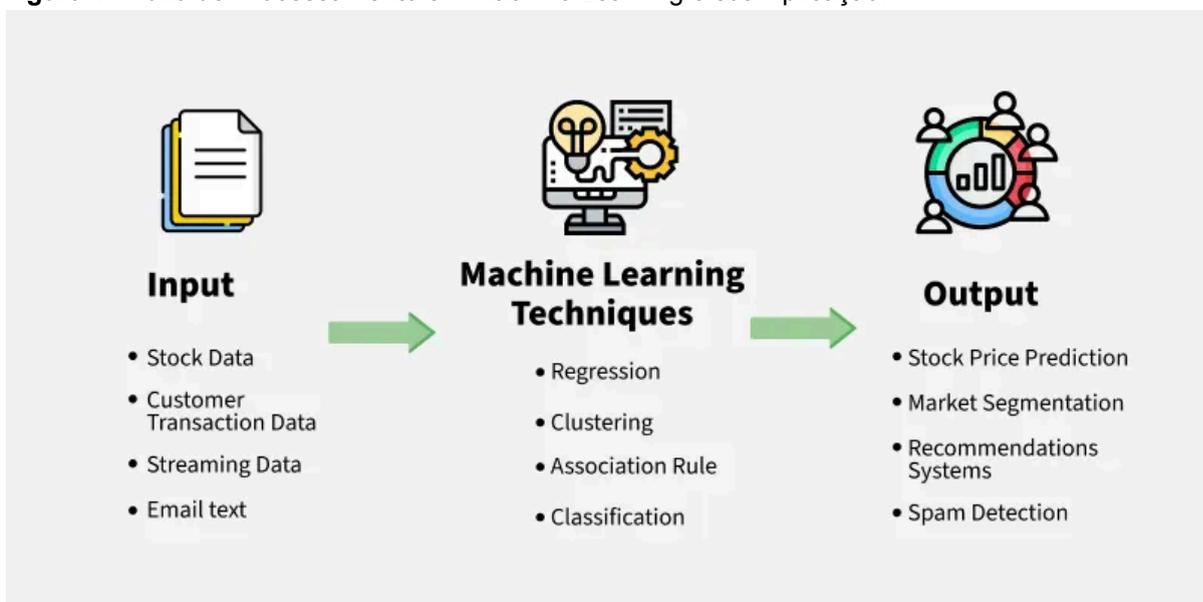
A inteligência artificial refere-se à capacidade de sistemas computacionais realizarem tarefas que normalmente exigiriam inteligência humana, como reconhecimento de padrões, tomada de decisão e previsão de eventos (AWS, 2023). Dentro da IA, o aprendizado de máquina (machine learning) se destaca como um dos principais métodos utilizados para permitir que sistemas computacionais aprendam a partir de dados, sem necessidade de programação explícita como foi demonstrado na Figura 1.

No desenvolvimento deste sistema semafórico inteligente, o aprendizado de máquina desempenhou um papel essencial, permitindo o treinamento da rede neural YOLOv11 com um dataset específico para a detecção de pedestres e suas respectivas condições de mobilidade com alta precisão.

O aprendizado de máquina tem sido amplamente aplicado em diversas áreas, incluindo reconhecimento de voz, atendimento ao cliente, recomendação de produtos e, mais relevante para este trabalho, visão computacional (IBM, 2023). A visão computacional é empregada em aplicações como etiquetagem automática de fotos em redes sociais, interpretação de imagens médicas para diagnósticos e detecção de objetos em tempo real para controle de tráfego (IBM, 2023). No setor automotivo, por exemplo, modelos de IA são usados para identificar veículos e pedestres em sistemas de direção autônoma, garantindo maior segurança e eficiência na tomada de decisões.

No contexto deste estudo, a implementação do YOLOv11 como ferramenta de visão computacional permite que o sistema semafórico identifique pedestres e ajuste o tempo de travessia de forma dinâmica. Essa abordagem possibilita a criação de um trânsito mais seguro e acessível, reduzindo a necessidade de intervenção manual e garantindo uma melhor adaptação ao fluxo real de pedestres. O uso de aprendizado de máquina na detecção de padrões visuais demonstra a viabilidade de tecnologias inteligentes aplicadas à mobilidade urbana, proporcionando benefícios tanto para pedestres quanto para motoristas.

**Figura 1:** Fluxo de Processamento em Machine Learning e sua Aplicação.



Fonte: (GeeksforGeeks, 2023).

A imagem ilustra um fluxo típico de processamento em Machine Learning, destacando as etapas principais envolvidas no desenvolvimento de modelos de aprendizado de máquina. Inicialmente, há a entrada de dados (*Input*), que pode incluir informações como transações de clientes, dados de mercado financeiro,

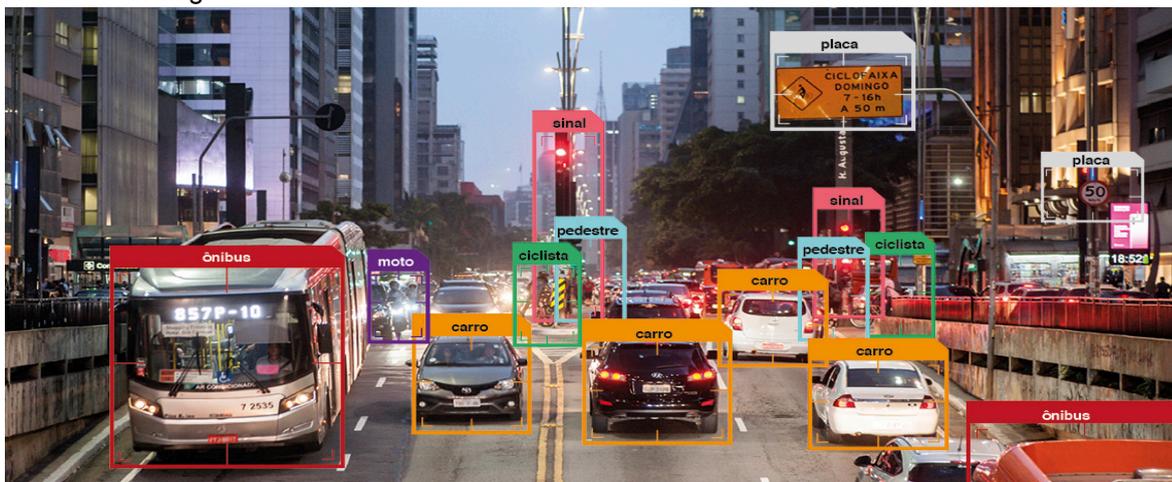
streaming de dados ou textos de e-mails. Em seguida, os dados passam por técnicas de aprendizado de máquina (*Machine Learning Techniques*), que envolvem métodos como regressão, agrupamento (*clustering*), regras de associação e classificação. Por fim, o sistema gera resultados (*Output*), aplicáveis a diversas áreas, como previsão de preços de ações, segmentação de mercado, sistemas de recomendação e detecção de spam. Esse processo é amplamente utilizado em diversas áreas, incluindo segurança, análise financeira e visão computacional, sendo um princípio fundamental também para o desenvolvimento do sistema semafórico inteligente proposto neste trabalho. No contexto deste projeto, a abordagem de aprendizado de máquina foi aplicada à visão computacional para detecção e classificação de pedestres, permitindo ajustes dinâmicos nos tempos semafóricos de acordo com a demanda real.

Dessa maneira, a implementação da Inteligência Artificial (IA) e do aprendizado de máquina desempenhou um papel fundamental no desenvolvimento do sistema, permitindo que ele adquirisse a capacidade de generalização. Isso significa que o modelo consegue identificar corretamente pedestres em diferentes cenários, mesmo quando confrontado com variações não presentes no conjunto de treinamento. Essa habilidade é essencial para garantir a robustez do sistema em um ambiente real, onde fatores como iluminação, ângulos de captura e vestimentas dos pedestres podem variar significativamente.

## **2.2 Visão Computacional**

A visão computacional é um campo da inteligência artificial que permite que computadores e sistemas extraiam informações relevantes a partir de imagens digitais, vídeos e outros insumos visuais, tomando decisões ou oferecendo recomendações com base nesses dados (IBM, 2023). Essa tecnologia é amplamente utilizada em diversas áreas, como segurança, medicina, indústria e transporte, proporcionando soluções avançadas para reconhecimento de padrões, detecção de objetos e tomada de decisões automatizadas. No contexto do trânsito urbano, a visão computacional tem sido aplicada para monitoramento de vias, controle de tráfego e, mais recentemente, para aprimorar a segurança dos pedestres em travessias, conforme a Figura 2.

**Figura 2:** Representação de um sistema que utiliza visão computacional para detecção de diferentes classes no tráfego urbano.



Fonte: (Chaves; Otubo, 2019).

Neste trabalho, a visão computacional é empregada para detectar pedestres em faixas semaforizadas, utilizando uma abordagem baseada em redes neurais convolucionais (CNNs) e modelos de detecção de objetos em tempo real. Através da captação contínua de imagens via câmera, o sistema é capaz de identificar a quantidade e o perfil dos pedestres, diferenciando indivíduos com mobilidade reduzida, como usuários de cadeiras de rodas, muletas e carrinhos de bebê, ajustando dinamicamente o tempo de travessia do semáforo com base na demanda real. Para alcançar essa funcionalidade, foi utilizada a arquitetura YOLOv11, um dos frameworks mais avançados para detecção de objetos em tempo real, que será discutido mais adiante.

### **2.3 Redes Neurais Convolucionais (CNNs)**

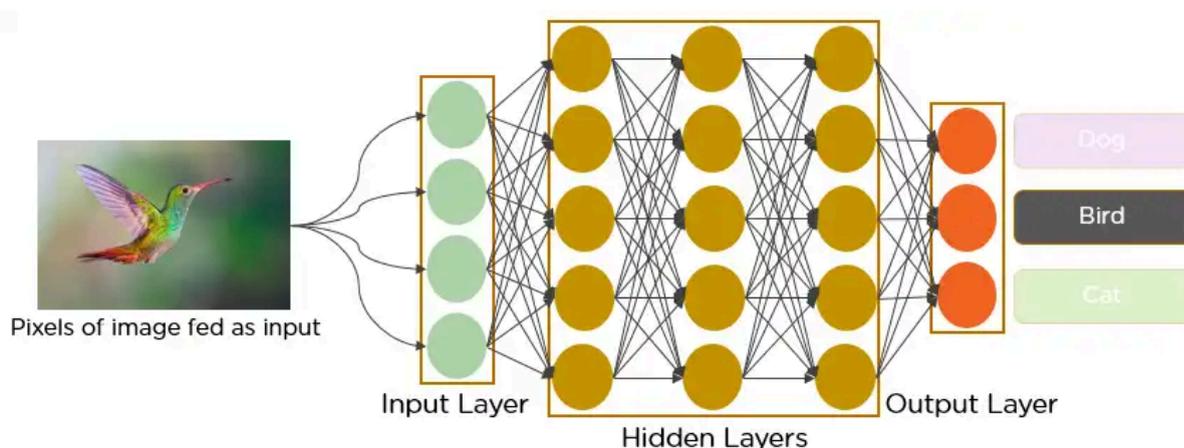
As Redes Neurais Convolucionais (Convolutional Neural Networks - CNNs) são um tipo de rede neural artificial otimizada para processamento de dados estruturados em forma de grade, como imagens (IBM, 2023). Elas se diferenciam das redes neurais tradicionais por possuírem camadas especializadas na extração de características visuais, permitindo que o modelo aprenda a identificar padrões complexos dentro de imagens.

No desenvolvimento do sistema semaforico inteligente, foi utilizado o modelo YOLOv11, uma rede neural convolucional (CNN) especializada na detecção de objetos em tempo real. Esse modelo se destaca por sua eficiência na identificação de pedestres, mesmo em condições adversas, como variações de iluminação e

oclusões parciais. A CNN emprega filtros convolucionais para extrair características essenciais das imagens, destacando bordas, formas e texturas, o que aprimora a precisão da detecção e contribui para um funcionamento mais confiável do sistema.

A estrutura de uma CNN é composta por múltiplas camadas, cada uma com uma função específica na extração de informações da imagem, como ilustrado na Figura 3.

**Figura 3:** Arquitetura de Rede Neural para Classificação de Imagens (CNN).



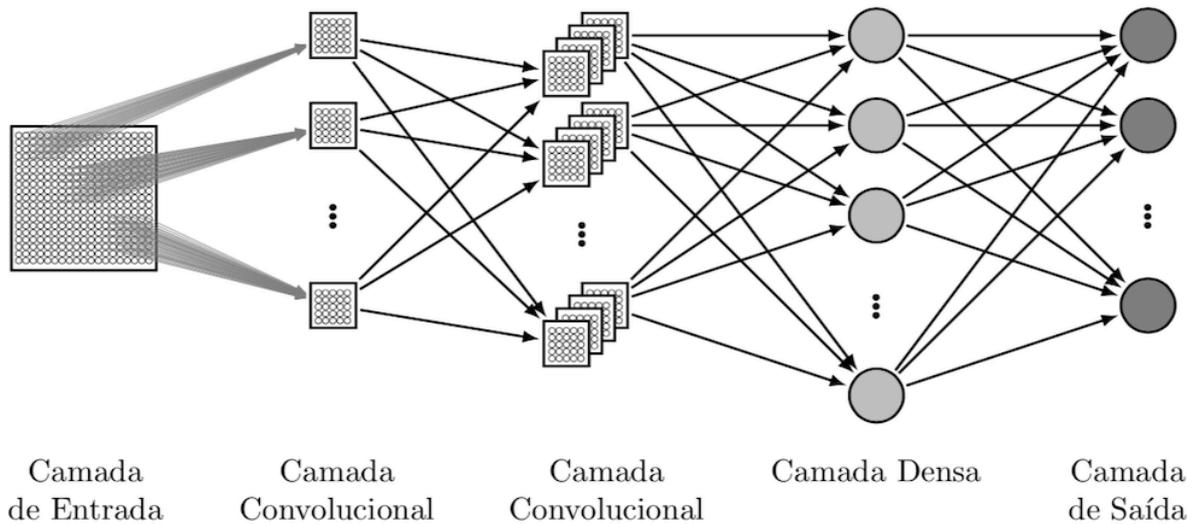
Fonte: (DIO, 2023).

### 2.3.1 Camada Convolutiva

A camada convolutiva é responsável por processar a imagem de entrada e extrair padrões relevantes através da aplicação de filtros matemáticos (kernels) (IBM, 2023). Durante essa etapa, cada filtro detecta diferentes características da imagem, como bordas, texturas e formas geométricas, permitindo que a rede identifique objetos com alta precisão, ilustrado no diagrama da Figura 4.

No contexto deste projeto, a camada convolutiva foi crucial para que o YOLOv11 aprendesse a diferenciar pedestres de outros elementos no ambiente urbano. Como resultado, o modelo conseguiu reconhecer indivíduos aguardando a travessia, distinguindo-os de objetos irrelevantes, como postes, placas ou veículos estacionados.

**Figura 4:** Diagrama de Fluxo de uma CNN para Processamento de Imagens.

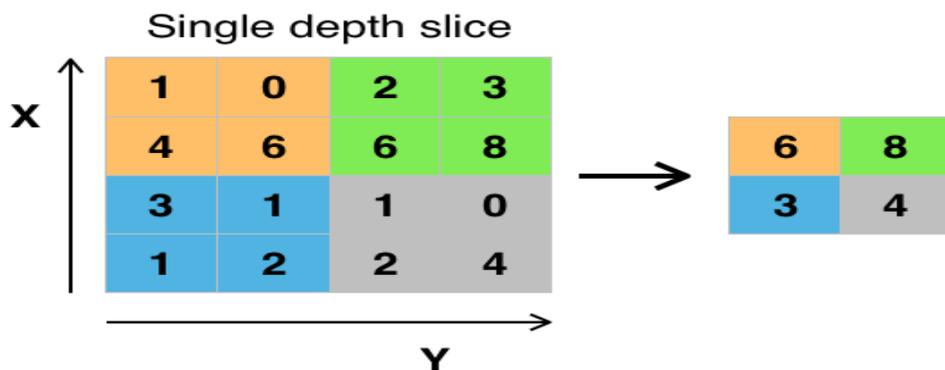


Fonte: (Sakurai, 2023).

### 2.3.2 Camada de Pooling

A camada de pooling tem o objetivo de reduzir a dimensionalidade dos mapas de características gerados pela convolução, mantendo apenas as informações mais relevantes (IBM, 2023). Isso ajuda a tornar a rede mais eficiente, reduzindo o tempo de processamento e o consumo de memória, sem perder a capacidade de generalização, demonstrado na Figura 5.

**Figura 5:** Operação de pooling em uma Rede Neural Convolucional (CNN).



Fonte: (DataCamp, 2023).

A Figura 5 demonstra a operação de max pooling aplicada a um mapa de características em uma Rede Neural Convolucional (CNN). A imagem apresenta uma matriz de entrada dividida em regiões menores, onde o maior valor de cada região é selecionado e propagado para a matriz de saída reduzida.

No projeto, essa camada permite que o modelo YOLOv11 processasse as imagens em tempo real, garantindo que a detecção dos pedestres fosse feita de forma ágil, sem comprometer a precisão.

#### **2.4 Frameworks de Visão Computacional: YOLOv11**

O YOLO (You Only Look Once) é um dos frameworks mais avançados e amplamente utilizados para detecção de objetos em tempo real. Sua arquitetura baseada em redes neurais convolucionais (CNNs) permite identificar múltiplos objetos em uma única passagem pela imagem, garantindo maior eficiência em comparação a abordagens tradicionais que utilizam múltiplas regiões de proposta para detecção (REDMON et al., 2016). Essa característica faz do YOLO uma solução ideal para aplicações que exigem processamento em tempo real, como monitoramento de tráfego, segurança pública, automação industrial e veículos autônomos (BOCHKOVSKIY et al., 2020).

A versão YOLOv11, desenvolvida pela Ultralytics, representa um avanço significativo em relação às versões anteriores, trazendo otimizações que impactam diretamente a acurácia e o desempenho computacional. Algumas das principais melhorias incluem:

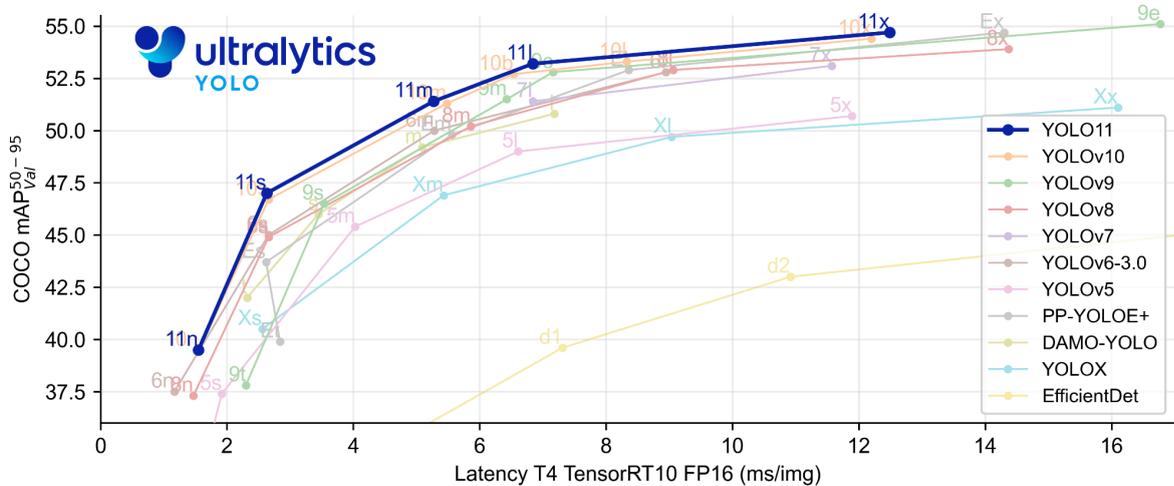
- Aprimoramento na precisão: A nova arquitetura da rede neural foi refinada para melhorar a identificação de objetos em diferentes condições ambientais, incluindo variações de iluminação e ângulos de câmera desfavoráveis.
- Eficiência computacional otimizada: O YOLOv11 foi projetado para maximizar a taxa de frames por segundo (FPS) sem comprometer a acurácia da detecção, o que o torna ideal para aplicações em tempo real.
- Detecção simultânea de múltiplos objetos: A nova versão melhora a segmentação e classificação de objetos, permitindo a identificação simultânea de pedestres, cadeiras de rodas, carrinhos de bebê e muletas com maior precisão.

A escolha do YOLOv11 para este projeto se deve à sua robustez e eficiência na detecção de pedestres em tempo real. Para um sistema de controle semafórico inteligente, é essencial que as informações sejam processadas com rapidez e precisão, garantindo a adaptação dinâmica dos tempos de travessia com base no fluxo de pedestres detectado. Além disso, a arquitetura baseada em CNNs permite

que a rede aprenda padrões específicos dos diferentes tipos de pedestres, aumentando a confiabilidade do sistema e reduzindo falsos positivos e negativos.

Na Figura 6, pode-se observar um exemplo da eficiência do YOLOv11 em comparação com as versões anteriores, demonstrando sua capacidade de generalização e adaptação a diferentes cenários. Essa característica foi fundamental para o desenvolvimento deste trabalho, pois possibilitou a implementação de um modelo robusto e escalável para aplicação em ambientes urbanos.

**Figura 6:** gráfico de desempenho comparativo entre diferentes versões do modelo YOLO.



Fonte: (Ultralytics, 2024).

### **3. TRABALHOS CORRELATOS**

O avanço das tecnologias de visão computacional e inteligência artificial tem impulsionado diversas pesquisas na área de monitoramento e controle do tráfego urbano. Diversos estudos vêm explorando a aplicação dessas tecnologias para otimizar a mobilidade urbana, melhorar a segurança viária e reduzir acidentes em travessias de pedestres. Trabalhos anteriores analisaram desde o controle dinâmico de semáforos baseado no fluxo veicular até a detecção de pedestres para ajustes automáticos no tempo de travessia.

Nesta seção, serão apresentados estudos correlatos que possuem semelhanças ou contém relações com este projeto, destacando suas abordagens, metodologias e contribuições para a área. Além disso, serão discutidas as principais diferenças entre essas pesquisas e a solução proposta neste trabalho, evidenciando a inovação e o diferencial da utilização do modelo YOLOv11 para a identificação de pedestres e o ajuste adaptativo do tempo semafórico com base no perfil dos indivíduos detectados.

#### **3.1 Metodologia para Controle de Semáforos Utilizando Processamento de Imagens para Reconhecimento da Quantidade de Carros**

O trabalho de Baroni (2018) propõe um sistema baseado em processamento de imagens e deep learning para reconhecer a quantidade de veículos em cruzamentos e otimizar o tempo dos semáforos de acordo com a demanda detectada. A abordagem utiliza a metodologia YOLO (You Only Look Once) para a detecção de veículos, integrando um modelo de inteligência artificial que ajusta os tempos semafóricos para reduzir congestionamentos. Esse estudo apresenta uma solução inovadora para a gestão do trânsito, voltada exclusivamente para a fluidez do tráfego de veículos, sem considerar aspectos relacionados à acessibilidade de pedestres.

#### **3.2 Controle de Semáforos em Tempo Real: A Experiência de São Paulo**

O estudo de Molist Vilanov (2005) explora a experiência da cidade de São Paulo com um sistema de controle de semáforos em tempo real, que ajusta os tempos semafóricos com base em sensores distribuídos pela cidade. O objetivo principal do sistema é reduzir congestionamentos, garantindo que vias com maior

demanda de veículos recebam mais tempo de sinal verde, enquanto aquelas com menor fluxo tenham sua abertura reduzida. A pesquisa foca na otimização do fluxo veicular, empregando sensores de tráfego tradicionais para o ajuste dinâmico dos semáforos.

### **3.3 Análise de Tráfego Urbano: Visão Computacional e Inteligência Artificial para Otimização de Semáforos**

O estudo de Vieira e Pereira (2023) apresenta um modelo de detecção e contagem automática de veículos baseado em visão computacional e computação de borda (*edge computing*). O sistema proposto utiliza câmeras e processamento digital de imagens para monitorar o fluxo veicular em tempo real, fornecendo uma solução de baixo custo para o gerenciamento do tráfego. A pesquisa demonstra a viabilidade da aplicação de inteligência artificial no monitoramento urbano, focando na contagem de veículos para otimização do tráfego, sem abordar adaptações específicas para outros perfis de usuários da via.

### **3.4 Uma Revisão Abrangente das Arquiteturas YOLO em Visão Computacional**

O estudo de Terven, Córdova-Esparza e Romero-González (2023) apresenta uma revisão completa da evolução da arquitetura YOLO, desde sua primeira versão até YOLOv8 e YOLO-NAS. O trabalho discute os avanços introduzidos em cada iteração, incluindo otimizações na estrutura de rede, aprimoramentos nas funções de perda e modificações na pós-processamento dos resultados. Além disso, são abordados os principais desafios enfrentados por esses modelos em aplicações do mundo real, como o balanceamento entre velocidade e precisão da detecção de objetos. O estudo destaca como o YOLO evoluiu para ser amplamente utilizado em diversas áreas, incluindo segurança, tráfego urbano e automação industrial, consolidando-se como uma das principais abordagens para detecção de objetos em tempo real.

### **3.5 Análise da Acessibilidade dos Espaços Urbanos para Usuários com Mobilidade Reduzida**

O estudo de Galindo (2017) analisa a acessibilidade urbana para pessoas com mobilidade reduzida, com foco nas calçadas e no transporte público na cidade

de São Paulo. A pesquisa avalia a conformidade da infraestrutura urbana com normas de acessibilidade e identifica desafios enfrentados por essa população em sua locomoção diária. Os resultados indicam que, apesar de avanços legislativos, a infraestrutura urbana ainda apresenta diversas barreiras que dificultam a mobilidade de pessoas com deficiência. O trabalho destaca a importância de considerar a acessibilidade no planejamento urbano, promovendo a inclusão e segurança desses indivíduos no espaço público.

### **3.6 Análise da Acessibilidade dos Espaços Urbanos para Usuários com Mobilidade Reduzida**

O trabalho de Barros (2021) propõe o uso de redes neurais convolucionais (CNNs) para a detecção de pedestres em ambientes urbanos, com o objetivo de melhorar a segurança viária e otimizar a mobilidade de pedestres. O estudo explora diferentes abordagens para treinamento de modelos de detecção, comparando a eficiência de arquiteturas baseadas em YOLO, Faster R-CNN e SSD. Os resultados mostram que o uso de CNNs pode melhorar significativamente a precisão da detecção de pedestres, reduzindo falsos positivos e aumentando a confiabilidade do sistema em ambientes reais. O estudo destaca ainda a importância da escolha do dataset e da técnica de treinamento para otimizar o desempenho dos modelos de visão computacional.

### **3.7 Correlação dos trabalhos**

A análise dos trabalhos existentes revela que a maioria das soluções voltadas para mobilidade urbana prioriza a fluidez do tráfego veicular, relegando a segurança dos pedestres a segundo plano. Estudos como os de Baroni (2018), Vilanov (2005) e Vieira e Pereira (2023) demonstram que a otimização do tempo semafórico tem sido majoritariamente voltada para reduzir congestionamentos e melhorar a circulação de veículos, utilizando sensores, processamento de imagens e redes neurais para monitoramento do tráfego. Entretanto, essas abordagens, embora eficientes para o fluxo viário, não consideram a adaptação do tempo semafórico para pedestres, especialmente aqueles com mobilidade reduzida.

Outros trabalhos, como o de Galindo (2017), trazem à tona a importância da acessibilidade no planejamento urbano, evidenciando que barreiras estruturais ainda

dificultam a locomoção segura de pessoas com deficiência. O estudo de Barros (2021) reforça que o uso de redes neurais convolucionais pode contribuir para a detecção mais precisa de pedestres, mas sua aplicação tem sido direcionada principalmente para a prevenção de colisões e veículos autônomos. Além disso, a revisão de Terven, Córdoba-Esparza e Romero-González (2023) destaca a evolução da arquitetura YOLO, evidenciando avanços significativos em precisão e desempenho, mas sem uma aplicação específica voltada à segurança de pedestres em travessias.

Diante desse panorama, o presente estudo se diferencia ao combinar visão computacional, redes neurais convolucionais e um modelo YOLOv11 customizado para criar um sistema inteligente de semáforo que prioriza pedestres de maneira adaptativa. Ao incorporar um mecanismo de pesos para ajustar o tempo de travessia, a solução proposta se destaca por propor garantir acessibilidade a pessoas com deficiência ou mobilidade reduzida, promovendo um trânsito mais seguro e inclusivo. Além disso, a implementação de uma abordagem em tempo real para análise e adaptação dinâmica do tempo semaforico representa um avanço na interseção entre tecnologia e mobilidade urbana, ampliando as possibilidades de integração de IA em infraestrutura pública. A tabela da figura 7 demonstra esse argumento:

**Figura 7:** Tabela-resumo da literatura analisada

Trabalhos	Uso de Visão Computacional	Detecção de Pedestres	adaptação do tempo semaforico	Propõe acessibilidade para usuários com mobilidade reduzida
Metodologia para Controle de Semáforos Utilizando Processamento de Imagens para Reconhecimento da Quantidade de Carros	X		X	
Controle de Semáforos em Tempo Real: A Experiência de São Paulo			X	
Análise de Tráfego Urbano: Visão Computacional e Inteligência Artificial para Otimização de Semáforos	X		X	
Uma Revisão Abrangente das Arquiteturas YOLO em Visão Computacional	X			
Análise da Acessibilidade dos Espaços Urbanos para Usuários com Mobilidade Reduzida				X
Análise da Acessibilidade dos Espaços Urbanos para Usuários		X		X

com Mobilidade Reduzida				
Autor	X	X	X	X

Fonte: Dados do autor.

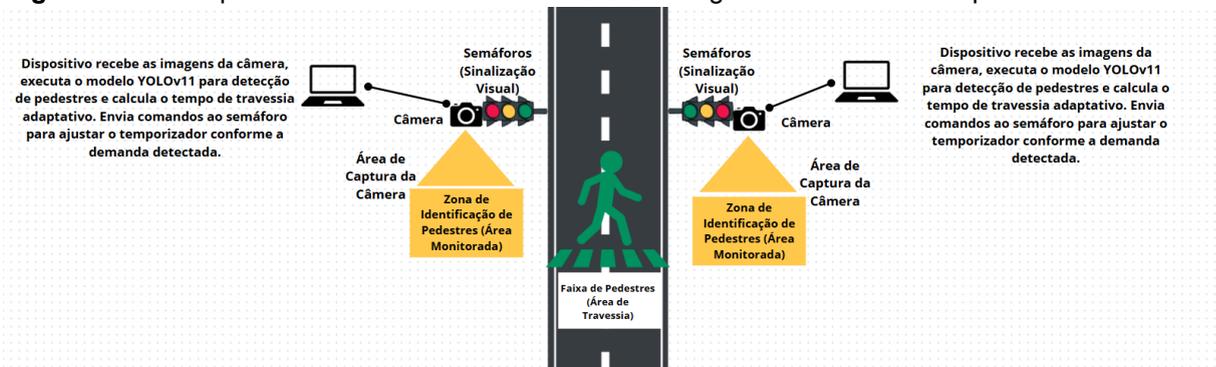
## 4. METODOLOGIA

A metodologia deste trabalho foi dividida em várias etapas, desde a pesquisa de campo até a implementação e validação do sistema proposto. Para garantir a precisão e confiabilidade dos resultados, o projeto foi conduzido seguindo etapas bem definidas, que incluem a coleta de dados sobre as condições reais das travessias, o treinamento de um modelo de detecção de pedestres utilizando YOLOv11, a realização de testes em diferentes mídias e a implementação de um sistema de simulação para validar o funcionamento da solução.

### 4.1 Sistema

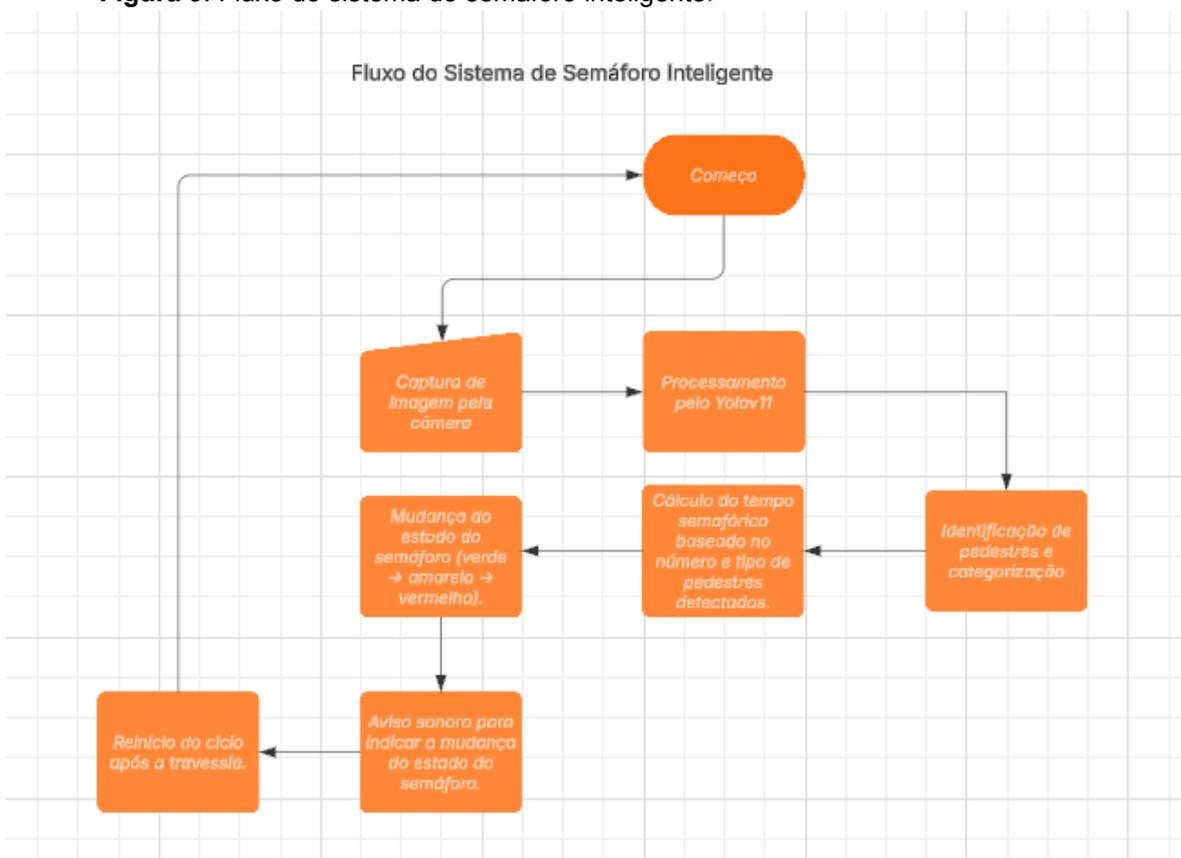
O processo se inicia com a captura de imagens por uma câmera posicionada próxima à faixa de pedestres. Essas imagens são processadas pelo modelo YOLOv11, que identifica e classifica os pedestres, diferenciando entre indivíduos comuns, cadeirantes, pessoas com muletas e carrinhos de bebê. Com base na quantidade e no tipo de pedestres detectados, o sistema calcula um tempo adequado para a abertura do semáforo, permitindo uma travessia segura, ilustrado na Figura 7 e Figura 8. Após a contagem regressiva, o semáforo altera seu estado (verde → amarelo → vermelho) e emite um aviso sonoro para alertar os pedestres e motoristas. O ciclo então se reinicia, aguardando a próxima detecção de pedestres para ajustar dinamicamente o tempo semafórico, ilustrado no diagrama da Figura 7 e Figura 8.

**Figura 8:** Fluxo Operacional do Sistema de Semáforo Inteligente com Visão Computacional.



Fonte: Dados do autor.

**Figura 9:** Fluxo do sistema do semáforo inteligente.



Fonte: Dados do autor.

## 4.2 Pesquisa de Campo

Antes da implementação do modelo de visão computacional, foi realizada uma pesquisa de campo para identificar problemas enfrentados pelos pedestres no momento da travessia em semáforos convencionais. A pesquisa foi aplicada a 20 pessoas, incluindo pedestres comuns e indivíduos com mobilidade reduzida. A pesquisa foi aplicada no semáforo localizado em Marabá/PA em frente ao Partage Shopping Marabá.

A pesquisa de campo foi conduzida com base em um questionário estruturado (Apêndice A) e obedeceu aos princípios éticos, garantindo o consentimento dos participantes por meio do Termo de Consentimento Livre e Esclarecido (Apêndice B) localizado no final do documento.

### 4.2.1 Anotação das Imagens (Rotularização das imagens)

A anotação das imagens é um processo essencial no treinamento de modelos de visão computacional, pois define as áreas de interesse dentro das imagens que o modelo deve aprender a reconhecer. No contexto deste trabalho, as imagens

utilizadas foram previamente rotuladas pelos criadores dos datasets disponibilizados no *Roboflow Universe*, uma plataforma que centraliza conjuntos de dados prontos para modelos de inteligência artificial.

A ferramenta Roboflow Annotation Tool foi utilizada para marcar manualmente as regiões das imagens onde pedestres, cadeiras de rodas, carrinhos de bebê e muletas estavam presentes. Esse processo consiste na criação de **bounding boxes** (caixas delimitadoras), que são retângulos desenhados ao redor dos objetos de interesse para indicar sua posição exata.

Para garantir a qualidade das anotações e evitar erros no treinamento do modelo, todas as marcações foram revisadas, assegurando que os rótulos estivessem uniformes e que cada objeto estivesse corretamente classificado. Esse cuidado foi essencial para evitar conflitos de indexação nos arquivos de *labels* (rótulos), o que poderia comprometer a precisão do modelo.

#### **4.2.2 Unificação do Dataset**

Após a anotação, foi realizada a padronização e unificação das imagens em um único conjunto de dados, utilizando um script na linguagem de programação Python, já que editar manualmente seria inviável pela quantidade de imagens/labels, e por fim, criando um arquivo `data.yaml` para treinar o modelo YOLOv11. Essa etapa foi essencial para garantir que todas as imagens seguissem um mesmo formato e que o modelo pudesse aprender a distinguir as classes pessoas, muletas e carrinho de bebê corretamente (já que em cadeiras de rodas houve overfitting).

#### **4.3 Desenvolvimento da Simulação do Sistema de Semáforo Inteligente**

Após a validação do modelo de detecção, foi desenvolvido um sistema de simulação de semáforo utilizando Python e OpenCV. A simulação consiste em o semáforo ficar aberto para veículos até que se tenha ao menos uma pessoa aguardando a travessia, após a identificação, a pessoa deve aguardar 30 segundos em frente ao semáforo para que o sistema contabilize e feche o sinal com o tempo estipulado. A interface gráfica foi projetada para exibir informações essenciais sobre o funcionamento do sistema, incluindo:

- O estado atual do semáforo (verde, amarelo ou vermelho).

- O tempo restante para a mudança de estado.

#### 4.4 Pesos e Parâmetros da simulação

O funcionamento da simulação segue uma série de regras definidas para o funcionamento do fluxo e garantir a alteração do temporizador para travessia.

Desse modo, utilizamos um tempo de travessia base (real e simulado), que é o tempo inicial para funcionamento da simulação do semáforo e tempo de espera (real e simulado) que é o tempo em que o pedestre deve aguardar para a travessia consiste em:

- Tempo base para travessia real: 30 segundos.
- Tempo base para travessia simulada: 30 segundos.
- Tempo base para passagem do tráfego veicular real: 1 minuto e 15 segundos.
- Tempo base para passagem do tráfego veicular simulado: 30 segundos.
- Tempo de identificação necessário para contabilizar pessoas: 30 segundos
- Tempo limite para não prejudicar o tráfego dos veículos: 1 minuto e 30 segundos

A partir disso, utilizamos dados da literatura científica para embasamento no cálculo, onde utilizo a fórmula cinemática - Movimento Retilíneo Uniforme (MRU) ( $S = S_0 + v \times t$ ) ou seja Tempo = Distância (m) / Velocidade (m/s). Utilizei esta fórmula para identificar o tempo médio de caminhada dos pedestres, considerando a média de caminhada de pessoas (KOVALYK, Poliana, 2023), arredondando os valores e considerando a velocidade de pessoas com idade avançada como base para criar-se a velocidade de pessoas com mobilidade reduzida em uma travessia média de 3 metros, criando os pesos de travessia, conforme as figuras 9 e 10.

**Figura 10:** Cálculo utilizando Movimento Retilíneo Uniforme para obter-se peso de tempo para pessoas sem mobilidade reduzida.

$$\frac{3m}{1,2m/s} \approx 2,5s$$

Fonte: Dados do autor.

**Figura 11:** Cálculo utilizando Movimento Retilíneo Uniforme para obter-se peso de tempo para pessoas com mobilidade reduzida.

$$\frac{3m}{0,6m/s} = 5s$$

Fonte: Dados do autor.

A partir dessas informações, utilizei os pesos 2 segundos para pessoas sem mobilidade reduzida e 5 segundos para pessoas com mobilidade reduzida. Criei uma fórmula heurística baseada em dados empíricos para o cálculo do tempo adaptativo, onde: Tempo Total de travessia = 30 segundos + (2 × quantidade de pessoas sem mobilidade reduzida) + (5 × quantidade de pessoas com mobilidade reduzida).

#### **4.7 Modularização do Código**

Para melhorar a organização, escalabilidade e manutenção do sistema, o código foi modularizado, ou seja, dividido em múltiplos arquivos, separando:

- Lógica de detecção e processamento de imagens
- Configuração do semáforo e tempos ajustáveis
- Sistema de áudio para alertas sonoros
- Exibição da interface gráfica com OpenCV

Essa abordagem torna o código mais organizado, permitindo ajustes fáceis e futuras implementações, além de demonstrar boas práticas de desenvolvimento no projeto.

#### **4.8 Sinal Sonoro e Acessibilidade**

Para garantir maior acessibilidade, foi implementado um sinal sonoro para indicar a mudança do semáforo. O sistema utiliza arquivos de áudio pré-gravados, que são acionados conforme a transição dos estados do semáforo, auxiliando pessoas com deficiência visual.

## 5. RESULTADOS E DISCUSSÃO

Nesta seção, são apresentados e analisados os principais resultados obtidos durante o desenvolvimento do sistema inteligente de controle semafórico. Primeiramente, é discutida a pesquisa de campo, que visou compreender a percepção dos pedestres em relação ao tempo de travessia e a necessidade de um sistema adaptativo. Em seguida, são detalhados os processos de construção do dataset e treinamento do modelo YOLOv11, destacando as métricas utilizadas para avaliar o desempenho da detecção de pedestres.

A pesquisa realizada com pedestres revelou dados significativos que validam a necessidade de um ajuste dinâmico do tempo semafórico. Os resultados demonstraram que grande parte dos entrevistados considera o tempo de travessia insuficiente e já presenciou situações de risco, como a necessidade de correr para concluir a travessia. Além disso, a maioria dos participantes acredita que um sistema automatizado poderia contribuir para a segurança e acessibilidade nas travessias.

Dando continuidade à análise, a construção do dataset customizado foi um fator determinante para o sucesso do modelo de visão computacional. Diferentemente de datasets padrão, como o COCO, o conjunto de dados utilizado neste trabalho foi adaptado para incluir classes específicas de pedestres, como cadeiras de rodas, muletas e carrinhos de bebê, permitindo uma detecção mais inclusiva.

Para avaliar a eficácia do modelo, foi conduzido um processo de treinamento utilizando diferentes variantes do YOLOv11. A versão YOLOv11m foi escolhida devido ao melhor equilíbrio entre desempenho e consumo de recursos computacionais. Durante o treinamento, diversas métricas foram analisadas, incluindo a Matriz de Confusão Normalizada, Curva F1 e Precision-Recall Curve, para validar a acurácia do modelo.

Além disso, foram realizados testes comparativos entre o modelo treinado e o COCO Dataset, evidenciando uma melhoria na detecção de pedestres quando utilizando o dataset customizado. A partir disso, foi possível validar que a adaptação do dataset contribuiu para um desempenho superior do modelo, possibilitando sua implementação em um ambiente de simulação.

Por fim, os resultados da simulação do semáforo inteligente foram analisados para demonstrar a aplicabilidade do modelo em um cenário real. O sistema implementado foi capaz de identificar pedestres e ajustar o tempo de travessia de

forma dinâmica, seguindo a lógica pré-definida. Esses resultados reforçam o potencial da solução proposta para aumentar a segurança dos pedestres e promover uma mobilidade urbana mais eficiente.

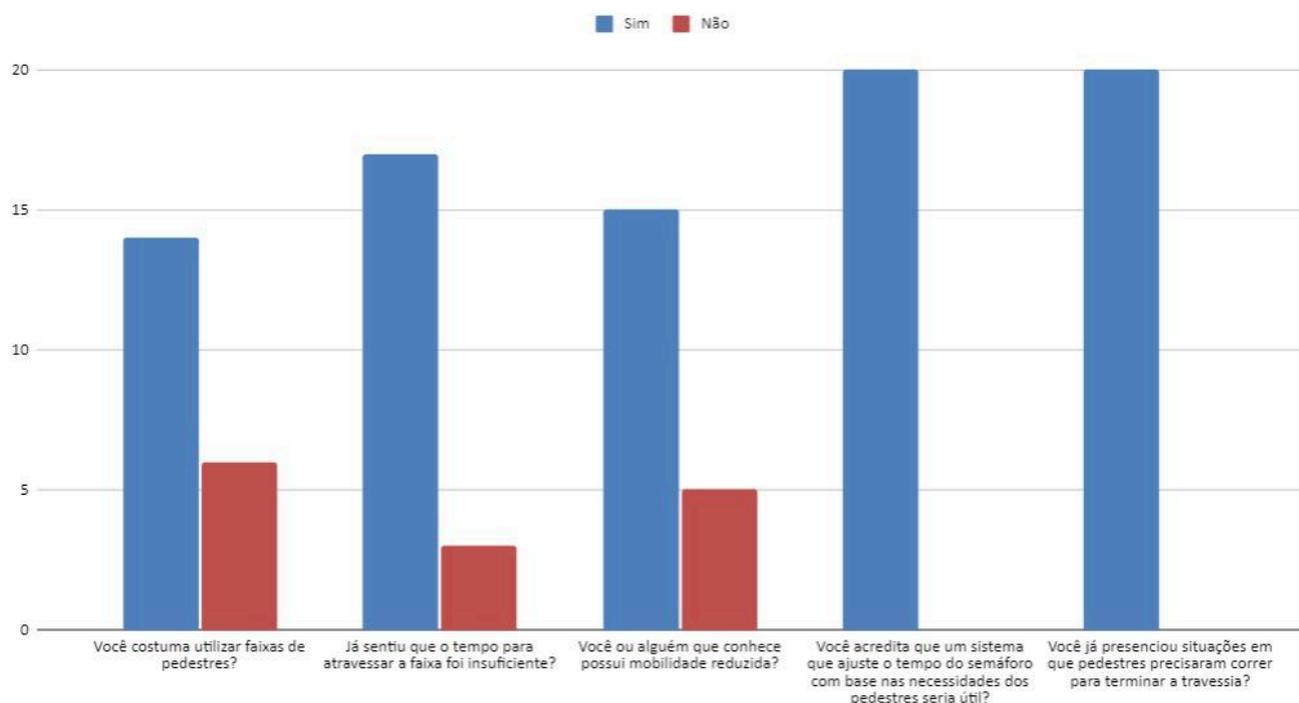
### **5.1 Pesquisa de Campo**

Foi realizada uma pesquisa de campo com os 20 entrevistados, dentre eles, pessoas portando carrinho de bebê e idosos (que são referenciais para pessoas com mobilidade reduzida) e pode ser observado na Figura 11. Obteve-se os seguintes resultados:

- 85% dos entrevistados afirmaram que o tempo de travessia é insuficiente, reforçando a necessidade de adaptação dos tempos semafóricos.
- 100% dos participantes já presenciaram pedestres correndo para atravessar a faixa antes do sinal fechar, evidenciando um risco significativo de acidentes.
- 75% dos entrevistados conhecem alguém com mobilidade reduzida, destacando a importância de um sistema adaptativo que leve em consideração pedestres com dificuldades motoras.
- 100% dos participantes acreditam que um sistema de ajuste automático do tempo do semáforo seria útil para aumentar a segurança dos pedestres.

**Figura 12:** Gráfico da pesquisa

Percepção dos Pedestres sobre a Travessia e Tempo do Semáforo – Pesquisa de Campo no Partage Shopping Marabá



Fonte: Dados do autor.

A Figura 12 mostra os resultados da pesquisa na confirmam a necessidade de um semáforo adaptativo que leve em consideração a quantidade de pedestres e suas condições de mobilidade. Isso se alinha com estudos que apontam a ineficiência dos tempos semafóricos fixos e a necessidade de soluções inteligentes para mitigar riscos (MOBILIZE, 2023). Esses resultados validam a relevância do problema e fornecem dados concretos para embasar a proposta deste trabalho.

### 5.1.1 Construção do Dataset

A construção de um dataset adequado é um fator determinante para o sucesso de modelos de aprendizado profundo em visão computacional. Para este projeto, foi utilizado um dataset customizado, uma vez que o COCO Dataset padrão não contém as classes específicas utilizadas no projeto como cadeira de rodas, carrinho de bebê e muletas. A curadoria e organização dos dados seguiram recomendações da literatura especializada, que apontam a importância de datasets diversificados, balanceados e anotados de forma precisa para garantir um bom desempenho do modelo (Shorten; Kandel, 2019).

O dataset foi compilado a partir de imagens rotuladas disponíveis no Roboflow, um repositório amplamente utilizado para a obtenção de bancos de imagens anotadas para treinamento de modelos de inteligência artificial. As classes definidas para este projeto foram:

- Pessoa (pedestre comum) (Roboflow, 2023)
- Cadeira de rodas (Roboflow, 2022)
- Muleta (Roboflow, 2023)
- Carrinho de bebê (Roboflow, 2023)

Essas categorias foram escolhidas com base em estudos que demonstram a importância da diferenciação de perfis de pedestres para sistemas adaptativos de trânsito (Fang et al., 2020). Segundo esses estudos, um sistema de controle semafórico eficiente deve considerar não apenas a presença de pedestres, mas também suas necessidades individuais, garantindo maior acessibilidade e segurança.

Para garantir um desempenho otimizado do modelo, foi adotada uma abordagem baseada em pré-processamento e aumento de dados (data augmentation), conforme recomendado por Goodfellow, Bengio e Courville (2016). Técnicas como rotação, espelhamento e variação de brilho foram utilizadas para aumentar a diversidade do dataset e reduzir o risco de overfitting, permitindo ao modelo aprender melhor os padrões dos diferentes perfis de pedestres.

O processo completo de treinamento levou aproximadamente 28 horas, utilizando uma GPU RTX 3070 de 8GB, com uma divisão dos dados da seguinte forma:

- Treinamento: 16.469 imagens (87.12%)
- Validação: 1.566 imagens (8.28%)
- Teste: 868 imagens (4.59%)

Essa distribuição acaba não seguindo práticas, pois o dataset foi coletado do Roboflow, o que não garante boa generalização, gerando viés de treinamento, podendo ser um dos principais motivos do overfitting na classe cadeiras de rodas, sendo um ponto crucial de melhoria para implementação em casos reais.

### 5.1.2 Treinamento do Modelo YOLOv11m - Métricas

O treinamento do modelo foi realizado utilizando o framework YOLOv11 da Ultralytics, sendo testadas três variantes para avaliar a melhor performance:

- YOLOv11n (nano)
- YOLOv11m (medium)
- YOLOv11x (extra large)

Foi selecionada a versão que apresentou melhor equilíbrio entre precisão e desempenho de acordo com a figura 12 retirada do site oficial da ultralytics para implementação no sistema final, no caso a YOLOv11m (medium), já que equilibra melhor a quantidade de recursos computacionais utilizados e a qualidade de detecção, sendo um fator custo x benefício para o projeto. O treinamento foi realizado por meio do VScode, utilizando uma GPU NVIDIA RTX 3070 para aceleração. No treinamento foram utilizadas 50 épocas (epochs).

**Figura 13:** Tabela de desempenho entre as versões do Yolo11.

Modelo	tamanho (pixéis)	mAPval 50-95	Velocidade CPU ONNX (ms)	Velocidade T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

Fonte: ULTRALYTICS (2025).

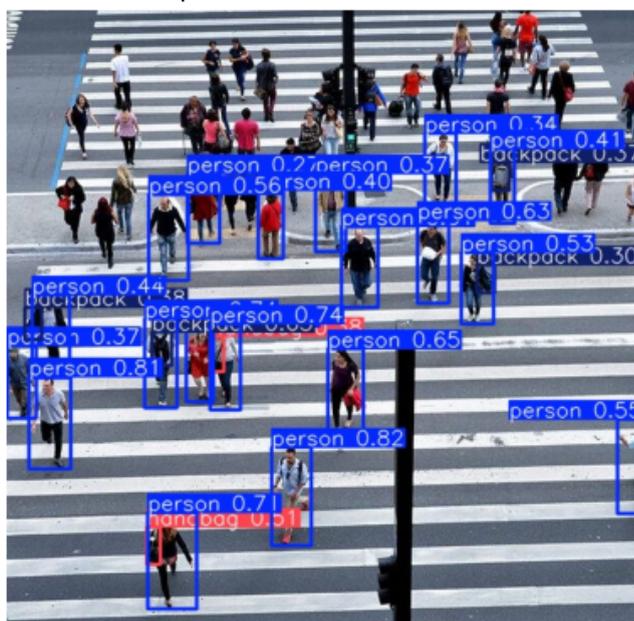
Durante o treinamento, o modelo gerou relatórios e métricas essenciais para avaliação, conforme descrito a seguir. Para testar o desempenho do modelo treinado, foram analisadas diversas métricas geradas pelo YOLOv11, incluindo:

- Confusion Matrix Normalized: Matriz de confusão normalizada, mostrando a precisão de detecção por classe.
- F1 Curve: Curva F1-score, usada para avaliar o equilíbrio entre precisão e recall.

Essas métricas foram fundamentais para a avaliação e otimização do treinamento do modelo, permitindo ajustes nos hiperparâmetros, como número de épocas (epochs) e tamanho do batch (batch size). Esses parâmetros influenciam

diretamente na capacidade do modelo de aprender padrões e realizar a detecção de pedestres com precisão. A análise da Matriz de Confusão Normalizada auxiliou na identificação de classes com menor acurácia, permitindo o balanceamento do dataset e ajustes no treinamento. Dessa forma, os ajustes baseados nessas métricas resultaram em uma melhora significativa na capacidade do modelo de identificar pedestres, superando a detecção feita pelo COCO Dataset, conforme demonstrado nas Figuras 14 e 15.

**Figura 14:** Total de 19 pessoas identificadas utilizando o cocodataset.



Fonte: Dados do autor.

**Figura 15:** Total de 51 pessoas identificadas utilizando o dataset customizado



Fonte: Dados do autor.

O modelo proposto foi construído conforme as seguintes etapas:

1. Estado Inicial: O semáforo inicia no verde para os veículos.
2. Detecção de Pedestres: Ao detectar pedestres aguardando na faixa, é iniciada uma contagem regressiva para a mudança do sinal.
3. Cálculo do Tempo de Travessia: 2 segundos adicionais para cada pedestre comum detectado e 5 segundos adicionais para cada pedestre com mobilidade reduzida detectado (cadeira de rodas, muletas ou carrinho de bebê).
4. Transição: O semáforo muda para amarelo por 2 segundos antes de ficar vermelho.
5. Fase Vermelha: Durante esse período, o tempo de travessia é ajustado com base no número de pedestres detectados.
6. Retorno ao Verde: Após a travessia dos pedestres, o semáforo retorna ao estado verde e o ciclo recomeça.

Podemos observar a lógica de funcionamento da simulação do semáforo (Apêndice C), os estados do semáforo nas Figuras 15, 16 e 17, as identificações das classes treinadas nas Figuras 18, 19 e 20 e a demonstração funcional nas Figuras 21 e 22.

**Figura 16:** Estado do semáforo contabilizando uma pessoa para a travessia.



Fonte: Dados do autor.

**Figura 17:** Estado do semáforo amarelo para representar atenção.



Fonte: Dados do autor

**Figura 18:** Estado do semáforo fechado, possibilitando a travessia dos pedestres.



Fonte: Dados do autor

**Figura 19:** Detecção da classe carrinho de bebê.



Fonte: Dados do autor

**Figura 20:** Detecção da classe muletas.



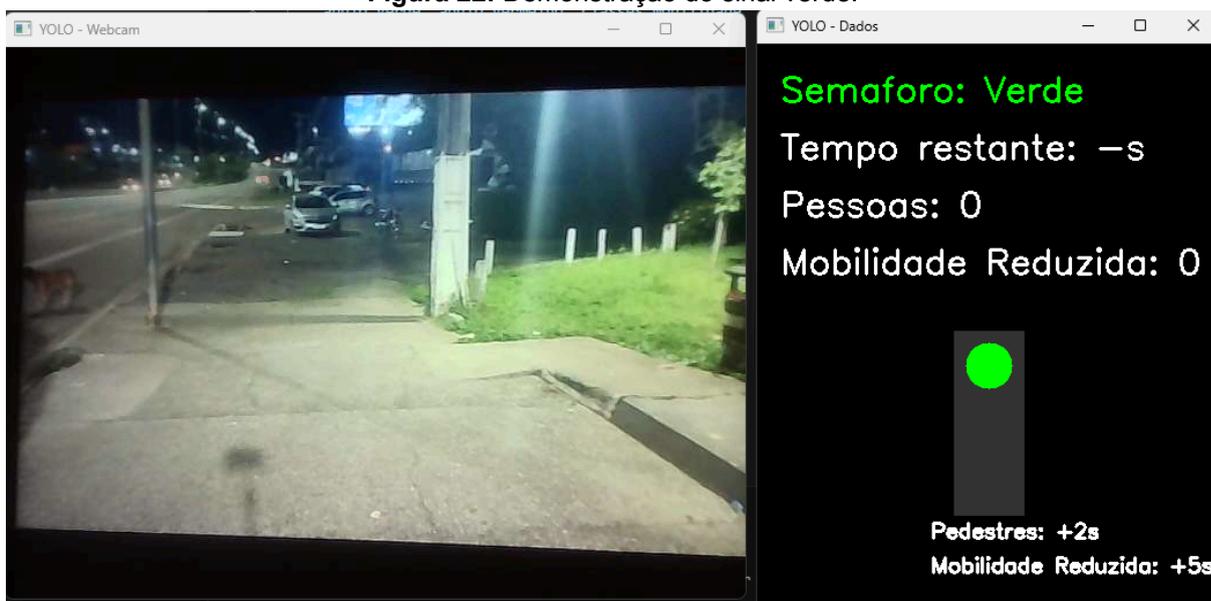
Fonte: Dados do autor

**Figura 21:** Má detecção da classe cadeira de rodas devido ao overfitting.



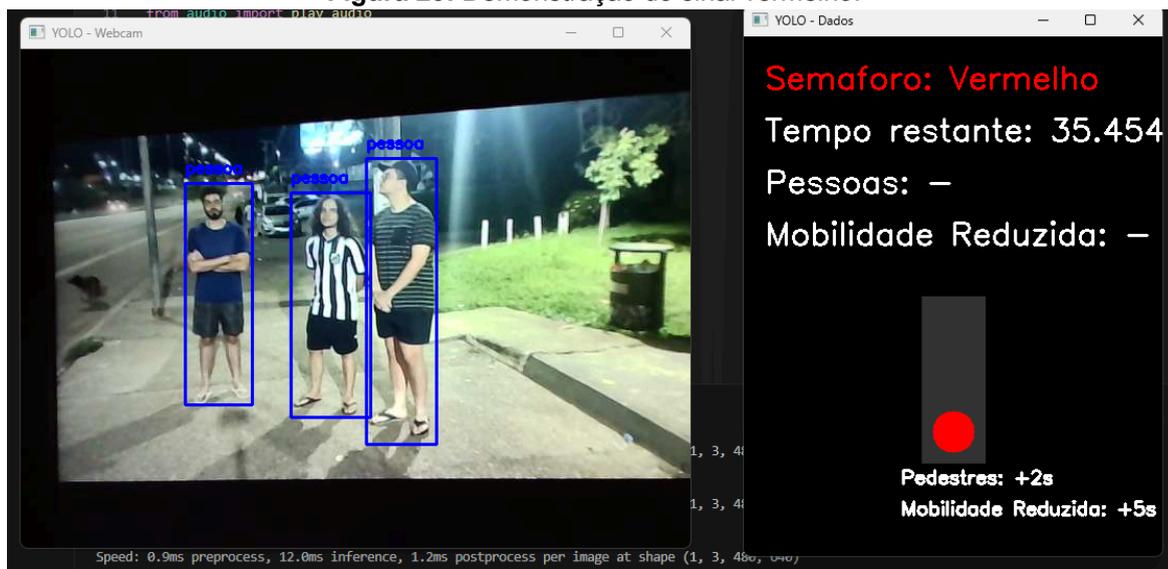
Fonte: Dados do autor

Figura 22: Demonstração do sinal verde.



Fonte: Dados do autor.

Figura 23: Demonstração do sinal vermelho.



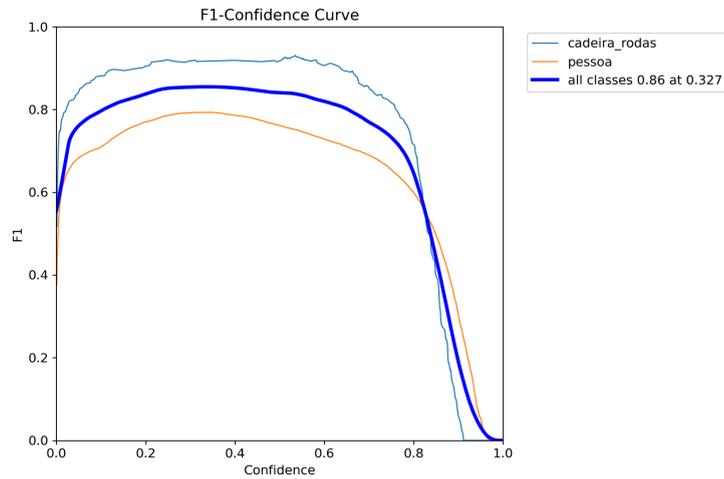
Fonte: Dados do autor.

Os testes evidenciaram que a classe cadeira de rodas não foi identificada com alta confiabilidade, possivelmente devido à baixa quantidade de imagens para validação e testes em comparação a pasta de treinamento. Além disso, o uso do Transfer Learning, que normalmente proporciona uma boa melhoria na acurácia, não teve o efeito esperado neste caso.

Isso ocorreu porque o modelo trata tudo dentro da *bounding box* como positivo ou negativo, e essa separação pode gerar perda de gradiente, reduzindo a capacidade do modelo de aprender corretamente os padrões da classe de cadeira de

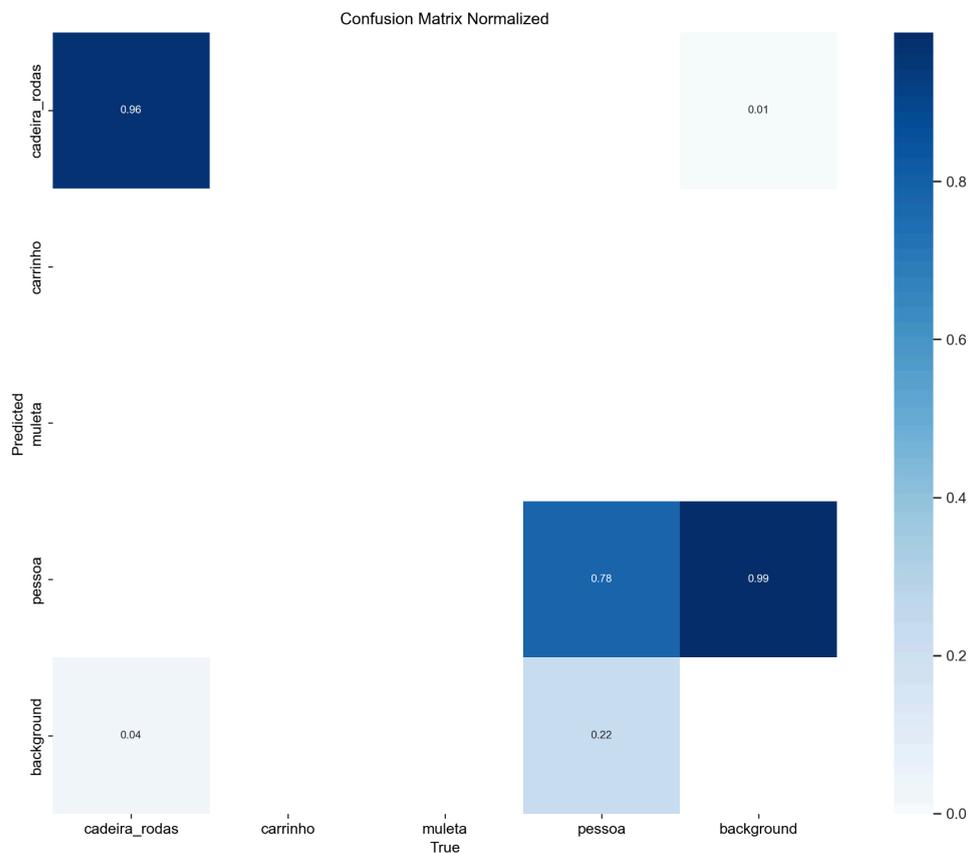
rodas. Com o tempo, o overfitting se torna elevado, e o modelo deixa de identificar essa classe de forma confiável. Podemos verificar os gráficos gerados após o treinamento nas Figuras 24 e 25.

**Figura 24:** Plot da matriz de confusão normalizada, mostrando a precisão de detecção por classe.



Fonte: Dados do autor.

**Figura 25:** Plot da matriz de confusão normalizada, mostrando a precisão de detecção por classe.



Fonte: Dados do autor

Os gráficos apresentados demonstram diferentes aspectos do desempenho do modelo treinado para a detecção de pedestres e identificação de indivíduos com dificuldades de locomoção em travessias de pedestres. O gráfico da matriz de confusão normalizada evidencia a taxa de acertos e erros do modelo para cada classe. No entanto, observa-se que as classes cadeira de bebê e muletas não aparecem no gráfico, indicando que o modelo não conseguiu aprender corretamente essas categorias. Esse problema pode estar relacionado à falta de dados suficientes no conjunto de treinamento ou ao viés na distribuição das amostras, resultando em uma baixa capacidade do modelo de reconhecer essas classes específicas.

Já o gráfico da F1 Confidence Curve ilustra a relação entre a confiança das previsões e o F1-score para diferentes limiares de detecção. Esse gráfico permite analisar o desempenho do modelo em diferentes níveis de confiança, auxiliando na escolha de um limiar ideal para otimizar a precisão e a revocação. No entanto, a curva também sugere uma inconsistência na detecção de certas classes, reforçando a necessidade de ajustes no treinamento ou refinamento dos dados utilizados.

Além disso, os resultados demonstram sinais de overfitting, um problema em que o modelo apresenta um alto desempenho nos dados de treinamento, mas um desempenho significativamente inferior nos dados de validação. Isso pode ser observado nas métricas de precisão e recall, que indicam que o modelo pode ter memorizado padrões específicos dos dados de treinamento, prejudicando sua capacidade de generalização. Esse comportamento pode ser mitigado com técnicas como aumento de dados (data augmentation), regularização (dropout, L2), ajuste da arquitetura da rede ou revisão do balanceamento do dataset.

Esses resultados reforçam a necessidade de ajustes no processo de treinamento para melhorar a robustez do modelo, garantindo que ele seja capaz de identificar corretamente todas as classes esperadas e operar de forma confiável em um ambiente real.

## **5.2 Testes**

Após a finalização do treinamento, o modelo foi testado de forma exploratória em diferentes tipos de mídia, buscando observar sua capacidade de identificar corretamente os objetos em cenários variados. Para essa análise qualitativa, foram utilizados exemplos diversos de entrada, permitindo verificar empiricamente o

comportamento do modelo em diferentes condições. Os testes foram realizados com os seguintes formatos de entrada:

- **Imagens estáticas:** Foram testadas imagens individuais de pedestres capturadas em ambientes urbanos reais. O objetivo era verificar se o modelo conseguia detectar corretamente cada classe treinada, avaliando sua precisão em identificar pedestres comuns, pessoas com carrinhos de bebê, usuários de muletas e cadeiras de rodas.
- **Vídeos gravados:** Foram utilizadas sequências de vídeo que simulam travessias reais em faixas de pedestres. Esse teste permitiu observar a consistência do modelo ao longo de múltiplos quadros, analisando sua capacidade de acompanhar o movimento dos pedestres e manter a precisão da detecção ao longo do tempo.
- **Transmissões ao vivo (lives):** Foi implementada uma captura de imagens em tempo real via webcam, recriando um cenário dinâmico semelhante ao funcionamento do sistema em um ambiente real. Esse teste foi essencial para validar a capacidade do modelo de operar continuamente sem perda significativa de desempenho, respondendo às mudanças no ambiente e detectando pedestres em tempo real.
- **Teste direto via webcam no computador:** Essa abordagem permitiu uma demonstração prática e acessível do funcionamento do modelo, facilitando a validação e ajustes em tempo real. Além de apresentar um bom desempenho na detecção das classes, essa foi a opção mais eficiente para a demonstração do sistema, uma vez que não dependia da disponibilidade de vídeos externos e possibilitava a replicação dos testes de forma simplificada e interativa.

Os testes exploratórios indicaram que o modelo foi capaz de identificar e classificar pedestres em diversos cenários, demonstrando um desempenho satisfatório em grande parte das situações observadas. A escolha de utilizar a webcam do computador para avaliação final foi motivada pela dificuldade em encontrar ou produzir vídeos que contemplassem todos os perfis de pedestres desejados. Além disso, essa abordagem permitiu uma análise mais dinâmica do comportamento do modelo, possibilitando observações diretas sobre sua

capacidade de detecção, bem como ajustes práticos com base nos resultados percebidos durante os testes.

### 5.3 Simulação do Sistema

Após o treinamento, foi desenvolvida uma simulação do semáforo para avaliar o funcionamento do modelo na prática. A interface gráfica foi implementada utilizando OpenCV podendo ser observados nas Figuras 26, 27 e 28, e exibindo:

- Estado do semáforo (verde, amarelo, vermelho);
- Número de pedestres detectados e tempo estimado para travessia;
- Cálculo de pesos para cada classe de pedestres;
- Sinal sonoro indicando mudança do semáforo (link de demonstração do projeto: <https://www.youtube.com/watch?v=w62TIAh5Ud0>).

**Figura 26:** Estado do semáforo contabilizando uma pessoa para a travessia.



Fonte: Dados do autor

**Figura 27:** Estado do semáforo amarelo para representar atenção.



Fonte: Dados do autor

**Figura 28:** Estado do semáforo fechado, possibilitando a travessia dos pedestres.



Fonte: Dados do autor

Conforme resultados das figuras **26**, **27** e **28** a simulação confirmou que o modelo funciona na prática, mas sua aplicabilidade em um ambiente real exigiria ajustes na robustez do modelo e integração com hardware de semáforos urbanos.

## **6. CONCLUSÃO**

### **6.1 Limitações e Trabalhos Futuros**

O modelo alcançou uma boa acurácia, mas longe do ideal. A identificação de cadeiras de rodas pode ser aprimorada, visto que teve problemas de overfitting. Como trabalho futuro, sugere-se a utilização de técnicas mais avançadas de Data Augmentation, como diversificação do dataset, melhor divisão das imagens (train, valid e test) utilizando como base boas práticas, além de implementação via hardware. Também é recomendável explorar diferentes abordagens de treinamento, como ajustes mais refinados no Transfer Learning ou mudança na estratégia de bounding boxes para minimizar a perda de gradiente. Com essas melhorias, espera-se um avanço na detecção de cadeiras de rodas e uma maior generalização do modelo para diferentes cenários.

### **6.2 Conclusão final**

Este trabalho apresentou o desenvolvimento e a validação de um sistema de semáforo inteligente baseado em visão computacional e aprendizado profundo, com o objetivo de tornar a travessia de pedestres mais segura e acessível. A metodologia adotada abrangeu desde a pesquisa de campo, que evidenciou a insatisfação dos pedestres com os tempos fixos dos semáforos, até o treinamento de um modelo YOLOv11m customizado, capaz de detectar diferentes perfis de pedestres e ajustar dinamicamente o tempo semafórico.

A pesquisa revelou que a maioria dos pedestres considera o tempo de travessia insuficiente, e que muitos já presenciaram situações de risco, como pessoas correndo para atravessar antes do sinal fechar. Além disso, grande parte dos entrevistados conhece alguém com mobilidade reduzida, reforçando a necessidade de um sistema que leve essas condições em consideração. Esses dados embasaram a proposta deste trabalho, garantindo que a solução desenvolvida fosse de fato relevante e aplicável à realidade urbana.

Com base nos testes realizados, foi possível confirmar que o modelo YOLOv11 obteve sucesso na detecção e classificação de pedestres, diferenciando indivíduos com mobilidade reduzida, como usuários de muletas, cadeiras de rodas e carrinhos de bebê. O sistema de adaptação do tempo semafórico também demonstrou eficácia ao calcular dinamicamente o tempo de travessia com base na

quantidade e no perfil dos pedestres detectados. No entanto, foi identificado um desempenho inferior na detecção de cadeiras de rodas, devido à baixa quantidade de imagens dessa classe no dataset de treinamento. Esse problema pode ser mitigado em estudos futuros com a inclusão de um conjunto de dados mais robustos.

A implementação do sistema foi realizada por meio de uma simulação, utilizando Python e OpenCV para representar visualmente o funcionamento do semáforo inteligente. Essa simulação permitiu demonstrar, de maneira prática, a adaptação do tempo semafórico em diferentes cenários, validando o funcionamento do sistema sem a necessidade de testes em ambiente real. Além disso, foi incorporado um sinal sonoro, garantindo maior acessibilidade para pedestres com deficiência visual.

Apesar dos resultados positivos, algumas limitações foram identificadas. A detecção de cadeiras de rodas apresentou um desempenho inferior em relação às demais classes, o que reforça a necessidade de aprimoramento do dataset para melhorar a precisão do modelo. Além disso, embora a simulação tenha permitido validar a proposta, a aplicação em um ambiente real ainda exigiria testes adicionais, considerando fatores como iluminação variável, movimentação desordenada de pedestres e diferentes condições climáticas.

Diante disso, algumas perspectivas para trabalhos futuros incluem o aprimoramento do dataset para tornar o modelo mais robusto, a realização de testes em ambiente real para validar sua aplicabilidade prática e a otimização do modelo para execução em hardware embarcado, permitindo sua integração a sistemas semafóricos já existentes. Além disso, a combinação desta abordagem com outras tecnologias, como sensores de tráfego e redes de comunicação veicular, poderia contribuir ainda mais para a construção de cidades inteligentes e um trânsito mais seguro e eficiente.

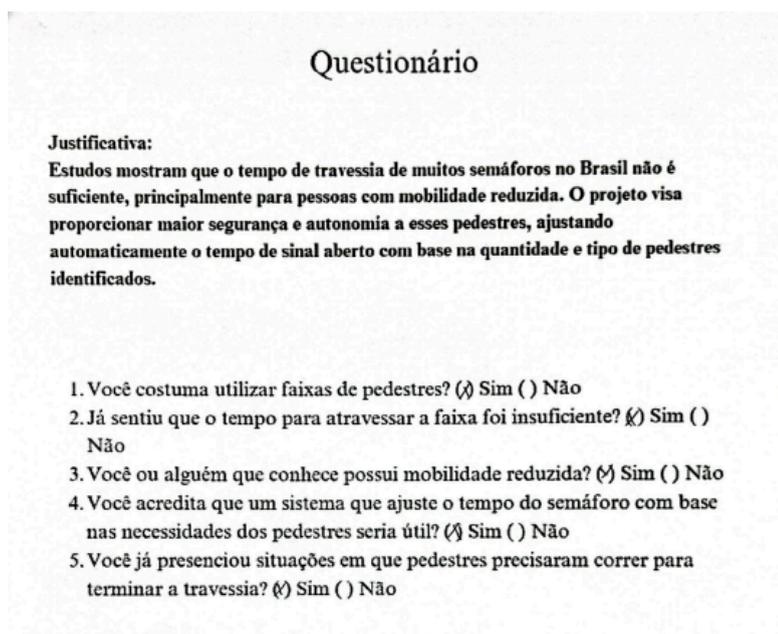
Em conclusão, o desenvolvimento deste semáforo inteligente adaptativo demonstrou-se uma alternativa viável e inovadora para otimizar a segurança de pedestres em travessias urbanas. A combinação de inteligência artificial, acessibilidade e adaptação dinâmica do tempo semafórico destaca este trabalho como uma contribuição relevante para a mobilidade urbana. Com a evolução das tecnologias de machine learning e visão computacional, soluções como esta podem se tornar cada vez mais comuns, promovendo um trânsito mais seguro, eficiente e inclusivo para todos.

## APÊNDICE A – QUESTIONÁRIO ESTRUTURADO

Neste apêndice, apresenta-se o questionário estruturado utilizado para a coleta de dados junto aos participantes da pesquisa. O questionário foi elaborado com o objetivo de compreender a percepção dos indivíduos sobre a segurança viária e a viabilidade da implementação de um semáforo inteligente baseado em visão computacional.

O instrumento de pesquisa contém perguntas fechadas e abertas, abordando temas como tempo de travessia, dificuldades enfrentadas por pedestres e possíveis melhorias no trânsito. A aplicação do questionário foi realizada de forma anônima, garantindo a confidencialidade das respostas.

**Figura 29:** Questionário final estruturado e apresentado para os participantes da pesquisa de campo.



**Questionário**

**Justificativa:**  
Estudos mostram que o tempo de travessia de muitos semáforos no Brasil não é suficiente, principalmente para pessoas com mobilidade reduzida. O projeto visa proporcionar maior segurança e autonomia a esses pedestres, ajustando automaticamente o tempo de sinal aberto com base na quantidade e tipo de pedestres identificados.

1. Você costuma utilizar faixas de pedestres?  Sim  Não
2. Já sentiu que o tempo para atravessar a faixa foi insuficiente?  Sim  Não
3. Você ou alguém que conhece possui mobilidade reduzida?  Sim  Não
4. Você acredita que um sistema que ajuste o tempo do semáforo com base nas necessidades dos pedestres seria útil?  Sim  Não
5. Você já presenciou situações em que pedestres precisaram correr para terminar a travessia?  Sim  Não

Fonte: Dados do autor.

## APÊNDICE B – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Este apêndice apresenta o Termo de Consentimento Livre e Esclarecido (TCLE), documento essencial para garantir que os participantes da pesquisa estivessem devidamente informados sobre seus objetivos, procedimentos e eventuais riscos e benefícios. O termo detalha o propósito do estudo, a metodologia adotada e o caráter voluntário da participação, assegurando que os respondentes pudessem se retirar da pesquisa a qualquer momento sem qualquer prejuízo. Além disso, o documento esclarece sobre a confidencialidade das informações fornecidas, garantindo que os dados coletados seriam utilizados exclusivamente para fins acadêmicos.

**Figura 30:** Termo de Consentimento Livre e Esclarecido (TCLE) utilizado para respaldar o processo de pesquisa.

**UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ — UNIFESSPA**  
**FACULDADE DE ENGENHARIA DA COMPUTAÇÃO - FEC**

**TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)**

Universidade Federal do Sul e Sudeste do Pará **Data: 01/03/2025**  
Faculdade de Engenharia da Computação  
Autor: Luis Guilherme Góis Matos  
Título do TCC: Sistema Inteligente de Semáforo com Visão Computacional para Ajuste de Tempo com Base nas Necessidades de Travessia

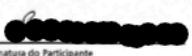
Prezado(a) participante,

Você está sendo convidado(a) a participar de uma pesquisa de campo que tem como objetivo coletar dados para o Trabalho de Conclusão de Curso (TCC) do curso de Engenharia da Computação da Universidade Federal do Sul e Sudeste do Pará. O projeto visa desenvolver um sistema inteligente de semáforo que utiliza visão computacional para ajustar o tempo de travessia de pedestres, levando em consideração as necessidades de pessoas com mobilidade reduzida.

A participação consiste em responder a um questionário e/ou ser observado(a) durante a travessia em faixas de pedestres, sem qualquer intervenção que ofereça riscos ou prejuízos à sua integridade física ou moral. As informações coletadas serão utilizadas exclusivamente para fins acadêmicos e serão tratadas com confidencialidade e anonimato, não sendo divulgados dados que permitam a sua identificação.

A sua participação é voluntária e você pode desistir a qualquer momento, sem prejuízo algum. Em caso de dúvidas ou necessidade de mais informações, você pode entrar em contato com o pesquisador responsável pelo projeto.

Declaro que fui devidamente informado(a) sobre os objetivos, procedimentos, riscos e benefícios da pesquisa, e que concordo em participar de forma livre e esclarecida.

X   
Assinatura do Participante

X   
Assinatura do Pesquisador

Fonte: Dados do autor.

## APÊNDICE C – LÓGICA DE FUNCIONAMENTO DA SIMULAÇÃO DO SEMÁFORO

A simulação do semáforo inteligente foi desenvolvida utilizando um algoritmo baseado em visão computacional para detectar pedestres e indivíduos com mobilidade reduzida. O sistema ajusta dinamicamente o tempo de travessia com base na presença e na quantidade de pedestres detectados.

### 1) Estados do Semáforo

- Verde: Permite a passagem de veículos. Durante esse período, a detecção de pedestres é ativada. Caso algum pedestre seja identificado, o tempo do sinal vermelho será ajustado.
- Amarelo: Fase de transição, alertando motoristas sobre a iminente mudança para o vermelho.
- Vermelho: O trânsito de veículos é interrompido, permitindo a travessia segura de pedestres. O tempo desse estado pode variar dependendo do número de pedestres detectados.

### 2) Detecção e Ajuste de Tempo

Quando o semáforo está verde, o sistema inicia a contagem de pedestres e de indivíduos com mobilidade reduzida utilizando a detecção de objetos com YOLO. Cada detecção é classificada e contabilizada:

- Caso algum pedestre seja identificado, inicia-se uma contagem do tempo de exposição ao verde.
- O tempo adicional do vermelho é calculado com base no número de pedestres e na presença de pessoas com mobilidade reduzida, respeitando um limite máximo predefinido.

Quando o tempo padrão do verde se esgota, o semáforo muda para amarelo e, logo depois, para vermelho, permanecendo nesse estado pelo tempo necessário para garantir a travessia segura. Após o tempo vermelho programado, o semáforo retorna ao verde, reiniciando o ciclo.

### 3) Exibição das Informações

Durante a simulação, um painel exibe informações como:

- Estado atual do semáforo
- Tempo restante para a troca de fase
- Número de pedestres detectados
- Número de indivíduos com mobilidade reduzida

Além disso, a interface visual apresenta um ícone do semáforo, alterando sua cor conforme o estado atual.

**Figura 31:** Código demonstrando lógica das condições para o funcionamento correto do semáforo.

```

if semaforo_estado == "Verde":
    num_pedestres = 0
    num_mobilidade = 0

    for i, box in enumerate(results[0].boxes.xyxy):
        class_id = int(results[0].boxes.cls[i])
        class_name = model.names[class_id]
        if class_name in classes_mobilidade:
            num_mobilidade += 1
        if class_id == 3:
            num_pedestres += 1

    if num_pedestres > 0 or num_mobilidade > 0:
        if green_timer_start is None:
            green_timer_start = time.time()
            adicional = (num_pedestres * tempo_por_pessoa) + (num_mobilidade * tempo_adicional_mobilidade)
            adicional = min(adicional, max_adicional)
            tempo_vermelho_atual = tempo_padrao_vermelho + adicional

            elapsed_green = time.time() - green_timer_start
            tempo_restante = max(0, tempo_padrao_verde - elapsed_green)
            if elapsed_green >= tempo_padrao_verde:
                semaforo_estado = "Amarelo"
                yellow_timer_start = time.time()
                green_timer_start = None
        else:
            green_timer_start = None
            tempo_restante = "-"

    elif semaforo_estado == "Amarelo":
        elapsed_yellow = time.time() - yellow_timer_start
        tempo_restante = max(0, yellow_duration - elapsed_yellow)
        if elapsed_yellow >= yellow_duration:
            semaforo_estado = "Vermelho"
            red_timer_start = time.time()
            yellow_timer_start = None

    elif semaforo_estado == "Vermelho":
        elapsed_red = time.time() - red_timer_start
        tempo_restante = max(0, tempo_vermelho_atual - elapsed_red)
        if elapsed_red >= tempo_vermelho_atual:
            semaforo_estado = "Verde"
            red_timer_start = None
            tempo_vermelho_atual = tempo_padrao_vermelho

```

Fonte: Dados do autor.

## REFERÊNCIAS

AWS. O que é rede neural? 2023. Disponível em:

<https://aws.amazon.com/pt/what-is/neural-network/>. Acesso em: 3 mar. 2025.

BARONI, R. B. Metodologia para Controle de Semáforos Utilizando Processamento de Imagens para Reconhecimento da Quantidade de Carros. 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Mecânica) – Universidade Tecnológica Federal do Paraná.

BARROS, João Henrique. Uso de Redes Neurais Convolucionais para Detecção de Pedestres em Ambientes Urbanos. 2021. Monografia – Universidade Federal de Pernambuco, Recife.

BARRA DO GARÇAS. Novo semáforo garantirá travessia segura de pedestres. 2023. Disponível em:

<https://www.barradogarcas.mt.gov.br/Imprensa/Noticias/Em-barra-do-garcas-novo-semaforo-garantira-travessia-segura-de-pedestres-4845>. Acesso em: 3 mar. 2025.

BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. Disponível em:

<https://doi.org/10.48550/arXiv.2004.10934>. Acesso em: 14 mar. 2025.

CARVALHO, M. R.; SOUZA, T. P. Segurança viária e tecnologia: desafios da mobilidade urbana inclusiva. Revista Brasileira de Mobilidade Urbana, v. 15, n. 3, p. 45-67, 2021.

CHAVES, Léo Ramos; OTUBO, Fabio. As máquinas que tudo veem. Revista Pesquisa FAPESP, São Paulo, 2019. Disponível em:

<https://revistapesquisa.fapesp.br/as-maquinas-que-tudo-veem/>. Acesso em: 3 mar. 2025.

DATA CAMP. Convolutional Neural Networks in Python. 2023. Disponível em:

<https://www.datacamp.com/pt/tutorial/convolutional-neural-networks-python>. Acesso em: 3 mar. 2025.

DIO. Redes neurais convolucionais (CNN): transformando dados em informação visual. 2023. Disponível em:

<https://www.dio.me/articles/redes-neurais-convolucionais-cnn-transformando-dados-em-informacao-visual>. Acesso em: 3 mar. 2025.

FANG, Z. et al. Pedestrian detection for intelligent transportation systems: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems, v. 21, n. 8, p. 3439-3452, 2020.

GALINDO, Priscila Marques. Análise da Acessibilidade dos Espaços Urbanos para Usuários com Mobilidade Reduzida. 2017. Monografia – Escola Politécnica da Universidade de São Paulo, São Paulo.

GEEKSFORGEES. ML | Machine Learning. 2023. Disponível em: <https://www.geeksforgeeks.org/ml-machine-learning/>. Acesso em: 3 mar. 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. MIT Press, 2016.

IBM. Casos de uso de aprendizado de máquina do mundo real. Disponível em: <https://www.ibm.com/br-pt/topics/machine-learning>. Acesso em: 3 mar. 2025.

IBM. O que é visão computacional? 2023. Disponível em: <https://www.ibm.com/br-pt/topics/computer-vision>. Acesso em: 3 mar. 2025.

IBM. O que é viés de IA? 2023. Disponível em: <https://www.ibm.com/br-pt/topics/ai-bias>. Acesso em: 3 mar. 2025.

IBM. Redes neurais convolucionais: entenda o funcionamento das CNNs. 2023. Disponível em: <https://www.ibm.com/br-pt/think/topics/convolutional-neural-networks>. Acesso em: 3 mar. 2025.

IBGE; MDHC. Brasil tem 18,6 milhões de pessoas com deficiência, indica pesquisa divulgada pelo IBGE e MDHC. Governo Federal, 07 jul. 2023. Disponível em: <https://www.gov.br/mdh/pt-br/assuntos/noticias/2023/julho/brasil-tem-18-6-milhoes-de-pessoas-com-deficiencia-indica-pesquisa-divulgada-pelo-ibge-e-mdhc>. Acesso em: 3 mar. 2025.

KOVALYK, Poliana. O que a velocidade que você caminha representa? Por que a velocidade da marcha é importante? Disponível em:

<https://guairaca.com.br/o-que-a-velocidade-que-voce-caminha-representa-por-que-a-velocidade-da-marcha-e-importante-por-hilana-martins>. Acesso em: 13 mar. 2025.

MOLIST VILANOV, L. O Controle de Semáforos em Tempo Real: A Experiência de São Paulo. Boletim Técnico CET, n. 38, 2005. Disponível em: <https://www.cetsp.com.br/consultas/publicacoes/boletins-tecnicos.aspx>. Acesso em: 15 set. 2024.

OBSERVATÓRIO DO TERCEIRO SETOR. SP: Semáforo deixa apenas seis segundos para pedestre atravessar. 2021. Disponível em: <https://observatorio3setor.org.br/sp-semaforo-deixa-apenas-seis-segundos-para-pedestre-atravesar/>. Acesso em: 3 mar. 2025.

ORGANIZAÇÃO MUNDIAL DA SAÚDE (OMS). Global status report on road safety 2023. Geneva: WHO, 2023. Disponível em: <https://www.transport-community.org/wp-content/uploads/2024/06/Global-status-report-on-road-safety-2023.pdf>. Acesso em: 3 mar. 2025.

REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. You Only Look Once: Unified, Real-Time Object Detection. 2016. Disponível em: <https://arxiv.org/abs/1506.02640>. Acesso em: 14 mar. 2025.

REVISTA FÓRUM TRANSPORTE. Utilização da IA no trânsito. 2023. Disponível em: <https://revistافت.com.br/utilizacao-da-ia-no-transito/>. Acesso em: 3 mar. 2025.

ROBOFLOW. Crutches Detection Dataset. 2023. Disponível em: <https://universe.roboflow.com/yul/cruches-detection>. Acesso em: 3 mar. 2025.

ROBOFLOW. People Detection Dataset. 2023. Disponível em: <https://universe.roboflow.com/leo-ueno/people-detection-o4rdr>. Acesso em: 3 mar. 2025.

ROBOFLOW. Strollers\_Only Dataset. 2023. Disponível em: [https://universe.roboflow.com/furnitureselectronics/strollers\\_only](https://universe.roboflow.com/furnitureselectronics/strollers_only). Acesso em: 3 mar. 2025.

ROBOFLOW. Wheel Chair Dataset. 2022. Disponível em: <https://universe.roboflow.com/hupi/wheel-chair>. Acesso em: 3 mar. 2025.

SAKURAI, G. CNN e MapReduce: um modelo de processamento para redes neurais convolucionais. 2023. Disponível em: <https://www.sakurai.dev.br/cnn-mapreduce/>. Acesso em: 3 mar. 2025.

SANTOS, L. F.; MELO, P. C.; FERREIRA, J. A. Tempo semafórico e segurança do pedestre: uma análise da acessibilidade urbana em grandes cidades. Revista de Engenharia de Tráfego, v. 12, n. 2, p. 89-104, 2020.

SHORTEN, C.; KANDEL, E. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, v. 6, n. 60, 2019.

TERVEN, Juan; CÓRDOVA-ESPARZA, Diana-Margarita; ROMERO-GONZÁLEZ, Julio-Alejandro. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning & Knowledge Extraction, v. 5, p. 1680–1716, 2023. DOI: 10.3390/make5040083.

ULTRALYTICS. Modelos YOLOv11. 2024. Disponível em: <https://docs.ultralytics.com/pt/models/yolo11/#overview>. Acesso em: 3 mar. 2025.