



UNIVERSIDADE FEDERAL DO MARANHÃO

Programa de Pós-Graduação em Ciência da Computação

Cícero Matias Ferreira do Nascimento Neto

Rastreamento de Veículos em Cidades Inteligentes por Meio de Sensoriamento Móvel Participativo

São Luís
2026

Cícero Matias Ferreira do Nascimento Neto

Rastreamento de Veículos em Cidades Inteligentes por Meio de Sensoriamento Móvel Participativo

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Ciência da Computação, ao Programa de Pós-Graduação em Ciência da Computação, da Universidade Federal do Maranhão.

Programa de Pós-Graduação em Ciência da Computação

Universidade Federal do Maranhão

Orientador: Prof. Dr. Francisco José da Silva e Silva

Coorientador: Prof. Dr. Luciano Reis Coutinho

São Luís - MA

2026

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Nascimento Neto, Cícero Matias Ferreira do.

Rastreamento de Veículos em Cidades Inteligentes por
Meio de Sensoriamento Móvel Participativo / Cícero Matias
Ferreira do Nascimento Neto. - 2026.

90 p.

Coorientador(a) 1: Luciano Reis Coutinho.

Orientador(a): Francisco José da Silva e Silva.

Dissertação (Mestrado) - Programa de Pós-graduação em
Ciência da Computação/ccet, Universidade Federal do
Maranhão, São Luís, Maranhão, 2026.

1. Rastreamento Veicular. 2. Cidades Inteligentes. 3.
Sensoriamento Móvel Participativo. 4. Computação de Borda.
5. Reconhecimento Automático de Placas Veiculares. I.
Silva, Francisco José da Silva e. II. Coutinho, Luciano
Reis. III. Título.

Cícero Matias Ferreira do Nascimento Neto

Rastreamento de Veículos em Cidades Inteligentes por Meio de Sensoriamento Móvel Participativo

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Ciência da Computação, ao Programa de Pós-Graduação em Ciência da Computação, da Universidade Federal do Maranhão.

Trabalho apresentado. São Luís - MA, 20 de Janeiro de 2026:

**Prof. Dr. Francisco José da Silva e
Silva**

Orientador

Universidade Federal do Maranhão

Prof. Dr. Luciano Reis Coutinho

Coorientador

Universidade Federal do Maranhão

Prof. Mario Antonio Meireles Teixeira

Examinador Interno

Universidade Federal do Maranhão

**Prof. Dr. Ricardo de Andrade Lira
Rabelo**

Examinador Externo

Universidade Federal do Piauí

São Luís - MA

2026

A Deus, fonte da vida e de toda a graça.

Agradecimentos

Agradeço primeiramente a Deus, fonte de vida, sabedoria e força, por conduzir cada etapa desta caminhada e tornar possível a conclusão deste trabalho.

Agradeço ao meu orientador, professor Francisco José da Silva e Silva, pela orientação atenta, pela disponibilidade e pelas contribuições fundamentais ao longo de todo o desenvolvimento desta pesquisa. Agradeço também ao professor Luciano Reis Coutinho, coorientador, pelas valiosas sugestões, apoio acadêmico e contribuições que enriqueceram este trabalho.

Aos colegas de laboratório, agradeço pela colaboração direta com esta linha de pesquisa, pelas discussões técnicas, trocas de conhecimento e pelo ambiente acadêmico colaborativo que contribuíram significativamente para a realização deste estudo.

Agradeço, de forma especial, aos meus familiares e amigos, pela paciência, apoio constante e pelas orações, que foram essenciais nos momentos de maior dedicação e esforço ao longo desta caminhada acadêmica.

Não te disse que, se creres, verás a glória de Deus?

Jo 11,40

Resumo

No contexto das Cidades Inteligentes, o rastreamento veicular desempenha papel fundamental em aplicações como a otimização da mobilidade urbana, a gestão do tráfego e, especialmente, o fortalecimento da segurança pública. Nesse cenário, a identificação e o acompanhamento de veículos roubados ou furtados constituem um desafio relevante, sobretudo em ambientes urbanos dinâmicos e com cobertura limitada de infraestrutura fixa. Este trabalho propõe um modelo de rastreamento veicular de baixo custo baseado em Sensoriamento Móvel Participativo, no qual vídeos do ambiente viário são capturados por smartphones posicionados em veículos ou em pontos estratégicos de fiscalização. O modelo integra técnicas de visão computacional e aprendizado profundo, abrangendo a detecção e o rastreamento de veículos, a extração das regiões de placas veiculares por meio do modelo YOLO e o reconhecimento dos caracteres utilizando o PaddleOCR. Para aumentar a confiabilidade da leitura, o sistema diferencia automaticamente entre o padrão Mercosul e o formato brasileiro descontinuado, aplicando regras específicas de validação para cada caso. Diferentes estratégias de processamento foram avaliadas experimentalmente, considerando tanto abordagens baseadas em processamento remoto quanto a execução local diretamente nos dispositivos móveis. Os resultados indicam que o processamento local apresenta desempenho mais consistente para a tarefa de rastreamento veicular, alcançando revocação média de 81,56% (*recall*, isto é, a taxa de identificação correta dos veículos efetivamente presentes no vídeo), além de maior estabilidade estatística quando comparado às abordagens dependentes de envio de dados. Como principal achado da pesquisa, observou-se que o processamento local se mostrou a estratégia mais adequada ao contexto avaliado, por apresentar maior estabilidade operacional, menor dependência de conectividade e viabilidade para uso contínuo em campo. Em contraste, as abordagens remotas mostraram-se mais sensíveis às condições da rede. Adicionalmente, essa estratégia operou dentro de limites aceitáveis de consumo de recursos computacionais, reforçando sua aplicabilidade prática. Os resultados evidenciam que o modelo proposto constitui uma alternativa robusta e escalável para aplicações de rastreamento veicular em Cidades Inteligentes, especialmente no apoio às ações de segurança pública, mesmo em cenários desafiadores de dupla mobilidade.

Palavras-chave: Rastreamento Veicular, Cidades Inteligentes, Sensoriamento Móvel Participativo, Computação de Borda, Reconhecimento Automático de Placas Veiculares, Segurança Pública.

Abstract

In the context of Smart Cities, vehicle tracking plays a fundamental role in applications such as urban mobility optimization, traffic management, and, especially, the enhancement of public safety. In this scenario, the identification and monitoring of stolen or stolen vehicles represent a significant challenge, particularly in dynamic urban environments with limited coverage of fixed infrastructure. This work proposes a low-cost vehicle tracking model based on Mobile Crowdsensing, in which videos of the road environment are captured by smartphones positioned in vehicles or at strategic inspection points. The model integrates computer vision and deep learning techniques, encompassing vehicle detection and tracking, extraction of license plate regions using the YOLO model, and character recognition using PaddleOCR. To increase reading reliability, the system automatically distinguishes between the Mercosur standard and the discontinued Brazilian format, applying specific validation rules for each case. Different processing strategies were experimentally evaluated, considering both remote processing approaches and local execution directly on mobile devices. The results indicate that local processing presents more consistent performance for the vehicle tracking task, achieving an average recall of 81.56% (recall, that is, the rate of correct identification of vehicles actually present in the video), in addition to greater statistical stability when compared to approaches dependent on data transmission. As the main finding of this research, local processing proved to be the most suitable strategy for the evaluated context, as it provides greater operational stability, lower dependence on connectivity, and feasibility for continuous field use. In contrast, remote approaches were more sensitive to network conditions. Additionally, this strategy operated within acceptable limits of computational resource consumption, reinforcing its practical applicability. The results demonstrate that the proposed model constitutes a robust and scalable alternative for vehicle tracking applications in Smart Cities, especially in supporting public safety actions, even in challenging dual mobility scenarios.

Keywords: Vehicle Tracking, Smart Cities, Mobile Participatory Sensing, Edge Computing, Automatic License Plate Recognition, Public Safety.

Lista de ilustrações

Figura 1 – Fluxo geral de uma aplicação de Sensoriamento Móvel Participativo, no qual dados sensoriais são coletados por dispositivos móveis, processados localmente e enviados ao <i>back-end</i> para agregação e mineração. Fonte: (GANTI; YE; LEI, 2011).	26
Figura 2 – Funcionamento do algoritmo <i>YOLO</i> . Fonte: (REDMON et al., 2016)	27
Figura 3 – Fonte: Adaptado de (CARDOSO, 2024)	39
Figura 4 – Ilustração do processo de marcação manual do veículo no sistema <i>CrowdTracking</i> , adaptado de Chen et al. (CHEN et al., 2019). (a) O veículo é marcado em duas fotografias capturadas pelos usuários. (b) O usuário toca na tela para selecionar manualmente o veículo. A imagem evidencia a necessidade de interação em tempo real por parte do participante, uma limitação significativa do método.	43
Figura 5 – Veículo equipado com câmera traseira. Fonte: (OHEKA; TU, 2018)	49
Figura 6 – Arquitetura do PNPR. Fonte: (ANANTHI et al., 2023)	50
Figura 7 – Estrutura do sistema LPR baseado no rastreamento de veículos e na integração de resultados. Fonte: (ZHU et al., 2019)	52
Figura 8 – Arquitetura Geral do <i>SmarTTrack</i> . Fonte: o autor	54
Figura 9 – SmarTTack Pipeline. Fonte: o autor	55
Figura 10 – (a) Aplicativo <i>SmarTTrack</i> utilizado em ambiente real no modo cidadão, em suporte veicular fixo. (b) Exemplos de placas veiculares extraídas automaticamente pelo sistema. Por questões de privacidade e proteção de dados, as informações alfanuméricas das placas foram propositalmente borradas nas imagens apresentadas. Fonte: o autor	56
Figura 11 – Matriz de confusão e métricas de desempenho do detector de placas baseado no <i>YOLOv11m</i> . Fonte: o autor	58
Figura 12 – Integração de resultados no <i>SmarTTrack</i> . Fonte: o autor	63
Figura 13 – Intervalo de confiança (IC 95%) das métricas de desempenho nos três cenários avaliados. Fonte: o autor	79
Figura 14 – Intervalo de confiança (IC 95%) do consumo de CPU nos três cenários. Fonte: o autor	80
Figura 15 – Intervalo de confiança (IC 95%) do consumo de memória nos três cenários. Fonte: o autor	80
Figura 16 – Comparativo do consumo médio de dados móveis nos três cenários avaliados. Fonte: o autor	81

Lista de tabelas

Tabela 1 – Comparação entre os sistemas analisados	53
Tabela 2 – Resultados de Desempenho para o Cenário 1(Local)	73
Tabela 3 – Resultados de Consumo de Recursos para o Cenário 1 (Local)	75
Tabela 4 – Resultados de Desempenho para o Cenário 2 (<i>Streaming</i>)	75
Tabela 5 – Resultados de Consumo de Recursos para o Cenário 2 (<i>Streaming</i>). . .	76
Tabela 6 – Resultados de Desempenho para o cenário 3 (blocos de vídeo).	77
Tabela 7 – Resultados de Consumo de Recursos para o Cenário 3 (Blocos de vídeo)	78

Lista de abreviaturas e siglas

ALPR	<i>Automatic License Plate Recognition</i>
API	<i>Application Programming Interface</i>
CCTV	<i>Closed-Circuit Television</i>
CNN	<i>Convolutional Neural Network</i>
CPU	<i>Central Processing Unit</i>
CRNN	<i>Convolutional Recurrent Neural Network</i>
CS	<i>Crowdsensing</i>
CV	<i>Computer Vision</i>
DB	<i>Differentiable Binarization</i>
FPS	<i>Frames Per Second</i>
GPU	<i>Graphics Processing Unit</i>
GPS	<i>Global Positioning System</i>
HSV	<i>Hue, Saturation, and Value</i>
IoT	<i>Internet of Things</i>
LPR	<i>License Plate Recognition</i>
LSTM	<i>Long Short-Term Memory</i>
MCS	<i>Mobile Crowdsensing</i>
ML	<i>Machine Learning</i>
OCR	<i>Optical Character Recognition</i>
REST	<i>Representational State Transfer</i>
SMP	<i>Sensoriamento Móvel Participativo</i>
YOLO	<i>You Only Look Once</i>

Sumário

1	INTRODUÇÃO	15
1.1	Contexto Geral	15
1.2	Caracterização do Problema	16
1.3	Justificativa	16
1.4	Visão Geral da Proposta	17
1.5	Objetivos	18
1.6	Contribuições	18
1.7	Organização do Trabalho	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	Cidades Inteligentes e Mobilidade Urbana	20
2.1.1	Elementos Estruturais das Cidades Inteligentes	22
2.2	Sensoriamento Participativo e Sensoriamento Móvel Participativo	22
2.3	Detecção de Objetos e o Algoritmo <i>YOLO</i>	26
2.4	Reconhecimento Automático de Placas Veiculares (ALPR)	28
2.5	Rastreamento de Múltiplos Objetos (MOT)	31
2.6	Reconhecimento Óptico de Caracteres	35
2.7	Integração com Plataformas Urbanas: A Arquitetura <i>InterSCity</i>	38
2.8	Considerações Finais	39
3	TRABALHOS RELACIONADOS	41
3.1	Metodologia para Seleção dos Trabalhos Relacionados	41
3.2	CrowdTracking: Real-Time Vehicle Tracking Through Mobile Crowdsensing	42
3.3	Smart Cities Mobility Monitoring Through Automatic License Plate Recognition and Vehicle Discrimination	44
3.4	Vehicle Route Tracking System Based on Vehicle Registration Number Recognition Using Template Matching Algorithm	46
3.5	Real-Time Multiple Vehicle Detection Using a Rear Camera Mounted on a Vehicle	47
3.6	Programmed Number Plate Recognition using Deep Learning	49
3.7	License Plate Recognition in Urban Road Based on Vehicle Tracking and Result Integration	51
3.8	Discussão Comparativa e Síntese	53
3.9	Considerações Finais	53

4	METODOLOGIA	54
4.1	Aquisição dos Dados	56
4.2	Treinamento e Validação do Detector de Placas	57
4.3	Deteccção de Veículos por meio de suas placas	59
4.4	Rastreamento de Veículos e Extração da Área da Placa	59
4.4.1	Rastreamento remoto	60
4.4.2	Rastreamento local	60
4.5	Definição do Padrão da Placa	61
4.6	Reconhecimento de Caracteres	62
4.7	Integração dos Resultados	63
4.8	Integração do <i>SmarTTrack</i> à Plataforma <i>InterSCity</i>	64
4.9	Considerações Finais	65
5	EXPERIMENTOS E RESULTADOS	67
5.1	Objetivo dos Experimentos	67
5.2	Materiais	68
5.3	<i>Dataset</i>	68
5.4	Métricas	68
5.5	Análise Estatística	69
5.6	Procedimentos e Métodos	71
5.7	Cenários Experimentais	71
5.8	Resultados	72
5.8.1	Cenário 1 – Processamento Local no Dispositivo	73
5.8.2	Cenário 2 — Processamento Remoto via <i>Streaming</i>	75
5.8.3	Cenário 3 — Processamento Remoto por Blocos de Vídeo	76
5.9	Comparação dos Cenários	78
5.9.1	Desempenho do Sistema	78
5.9.2	Consumo de CPU	79
5.9.3	Consumo de Memória	79
5.9.4	Consumo de Dados Móveis e Bateria	80
5.9.5	Considerações Finais	81
6	CONCLUSÕES	83
6.1	Contribuições	84
6.2	Limitações	84
6.3	Trabalhos Futuros	85
6.4	Publicações	86
	REFERÊNCIAS	87

1 Introdução

1.1 Contexto Geral

O acelerado avanço das Tecnologias da Informação e Comunicação (TIC) tem impulsionado o desenvolvimento das chamadas Cidades Inteligentes (*Smart Cities*), caracterizadas pela integração entre infraestrutura urbana, dispositivos conectados e serviços digitais com o propósito de aprimorar a eficiência operacional e a qualidade de vida da população. Nesse cenário, a mobilidade urbana assume papel central, exigindo mecanismos capazes de monitorar, compreender e responder de maneira dinâmica às condições do tráfego, à circulação de veículos e às demandas de segurança pública.

O rastreamento veicular, nesse contexto, surge como ferramenta estratégica tanto para fins de planejamento urbano quanto para aplicações de segurança, permitindo a identificação de veículos em situações de risco, acidentes, atividades suspeitas ou envolvimento em crimes. Tradicionalmente, tais sistemas são implementados por meio de câmeras fixas distribuídas em pontos estratégicos da cidade. Apesar de amplamente difundidas, essas soluções apresentam limitações relevantes: alto custo de implantação e manutenção, cobertura física restrita e vulnerabilidade a falhas operacionais.

Diante desse cenário, abordagens baseadas em Sensoriamento Participativo, especialmente no paradigma de Sensoriamento Móvel Participativo (SMP), têm ganhado destaque como alternativas escaláveis e de baixo custo. Nesse modelo, cidadãos e agentes públicos utilizam seus dispositivos móveis — como smartphones instalados em veículos — para capturar e enviar dados que subsidiam aplicações de monitoramento urbano. Essa forma de participação distribui a coleta de dados no espaço e no tempo, reduzindo a dependência de infraestrutura dedicada e ampliando significativamente a cobertura da malha de monitoramento (CHEN et al., 2019). Sistemas de visão computacional embutidos nesses dispositivos, aliados ao uso de algoritmos modernos de detecção baseados em *deep learning*, possibilitam que o processo de rastreamento aconteça de forma automatizada, transparente para o usuário e em tempo real (SPANU et al., 2021).

Além disso, a integração desses sistemas com plataformas urbanas de larga escala é essencial para viabilizar aplicações completas em Cidades Inteligentes. O *InterSCity*¹, plataforma aberta, escalável e baseada em microserviços, foi concebido justamente para apoiar aplicações urbanas distribuídas, oferecendo interfaces de gerenciamento, persistência e consulta de recursos heterogêneos (ESPOSTE et al., 2017). Ao integrar soluções participativas de rastreamento veicular ao *InterSCity*, torna-se possível registrar

¹ Projeto *InterSCity*: <<https://interscity.org>>

e disponibilizar informações de forma padronizada e interoperável, permitindo que agentes públicos consultem, em tempo real ou posteriormente, dados provenientes de múltiplos sensores móveis e fixos. Essa integração agrega robustez ao ecossistema urbano e potencializa o impacto de iniciativas baseadas em Sensoriamento Móvel Participativo.

1.2 Caracterização do Problema

O rastreamento veicular em Cidades Inteligentes exige mais do que a simples detecção de veículos ou a leitura isolada de placas. Em aplicações voltadas à segurança pública, é necessário executar um *pipeline* capaz de detectar, rastrear e reconhecer placas veiculares de forma contínua e confiável em ambientes urbanos reais, marcados por oclusões, variações de iluminação, trepidações e limitações de cobertura por infraestrutura fixa (CALITZ; HILL, 2020; ZHU et al., 2019; CHEN et al., 2019).

Tal questão torna-se ainda mais desafiadora em cenários de *dupla mobilidade*, nos quais tanto o veículo monitorado quanto o dispositivo de captura encontram-se em movimento simultâneo. Nessa condição, não basta alcançar boa acurácia em etapas isoladas de detecção ou ALPR; é necessário viabilizar a execução integrada do processo sob restrições de processamento, consumo de recursos e conectividade. Além disso, diferentes arquiteturas de execução — como processamento local no dispositivo ou envio dos dados para processamento remoto — podem afetar de maneira significativa o desempenho preditivo, a estabilidade operacional e a viabilidade prática da solução (CHEN et al., 2021; ZHU et al., 2019).

Dessa forma, o problema central desta pesquisa consiste em investigar como executar um *pipeline* de visão computacional para rastreamento veicular em cenários de dupla mobilidade e identificar qual arquitetura de processamento oferece o melhor equilíbrio entre desempenho preditivo, consumo de recursos e independência de conectividade. Essa lacuna motiva o desenvolvimento e a avaliação do *SmarTTrack* como alternativa para apoio a aplicações de segurança pública em Cidades Inteligentes.

1.3 Justificativa

O cenário da violência urbana no Brasil reforça ainda mais a relevância de sistemas capazes de identificar e rastrear veículos em tempo real. Dados do *Anuário Brasileiro de Segurança Pública* mostram que o país registra, de forma consistente, centenas de milhares de ocorrências de roubo e furto de veículos todos os anos. Em 2022, foram contabilizados 373.225 casos de subtração de veículos, somando roubos e furtos (FBSP, 2023). No ano seguinte, mesmo com leve redução, o número permaneceu elevado, totalizando 354.742 ocorrências em 2023 (FBSP, 2024). Em 2024, a tendência de queda prosseguiu,

alcançando ainda 344.596 registros (FBSP, 2025), evidenciando a persistência do problema em escala nacional. Esses dados demonstram que, apesar das oscilações anuais, o crime veicular mantém-se como um dos principais desafios da segurança pública brasileira, demandando soluções tecnológicas escaláveis, interoperáveis e de baixo custo, capazes de apoiar investigações e aumentar a capacidade de resposta das instituições policiais.

Outro aspecto relevante apontado pelo *Anuário Brasileiro de Segurança Pública* diz respeito às dinâmicas criminais associadas à subtração de veículos. Estudos presentes no relatório destacam que o roubo e o furto de automóveis alimentam mercados ilegais e cadeias de receptação estruturadas, frequentemente vinculadas a organizações criminosas (FBSP, 2023). Nessas dinâmicas, veículos subtraídos podem permanecer por longos períodos em circulação sem recuperação imediata, sendo utilizados como suporte para a prática de outros delitos, como assaltos, roubos a estabelecimentos ou ações de fuga. A dificuldade de rastrear veículos em tempo real, somada à infraestrutura limitada de monitoramento urbano, contribui para que muitos automóveis sejam abandonados pelos criminosos apenas após dias de uso ilícito, reforçando a necessidade de tecnologias que ampliem a capacidade investigativa e a velocidade de resposta das forças de segurança.

1.4 Visão Geral da Proposta

Como resposta ao problema apresentado, esta dissertação desenvolve o *SmarTTrack*, um modelo de rastreamento veicular baseado em Sensoriamento Móvel Participativo, voltado ao apoio a aplicações de segurança pública em Cidades Inteligentes. A proposta utiliza dispositivos móveis como elementos de captura e processamento, permitindo a obtenção de vídeos do ambiente viário por meio de *smartphones* posicionados em veículos ou em pontos estratégicos de fiscalização. A partir dessas capturas, o sistema executa um *pipeline* integrado de detecção de veículos, rastreamento múltiplo, extração de placas e reconhecimento automático de caracteres, com posterior registro seletivo dos eventos de interesse em uma plataforma de Cidades Inteligentes.

A arquitetura do *SmarTTrack* foi concebida para operar em cenários de dupla mobilidade. Para isso, a proposta combina técnicas de visão computacional e aprendizado profundo, integrando detecção baseada em *YOLO*, rastreamento, OCR e mecanismos de validação sintática das placas reconhecidas, considerando tanto o padrão Mercosul quanto o modelo brasileiro anterior. Além disso, o trabalho avalia diferentes estratégias de execução do sistema, comparando processamento local e remoto com o objetivo de identificar a alternativa mais adequada em termos de desempenho preditivo, consumo de recursos e dependência de conectividade.

Com essa abordagem, busca-se reduzir a dependência exclusiva de câmeras fixas e ampliar as possibilidades de monitoramento urbano colaborativo, oferecendo uma solução de

baixo custo, distribuída e integrada ao ecossistema computacional das Cidades Inteligentes.

1.5 Objetivos

O objetivo geral deste trabalho consiste em desenvolver e avaliar um modelo de rastreamento múltiplo de veículos baseado em Sensoriamento Móvel Participativo para uso em Cidades Inteligentes, capaz de realizar detecção, rastreamento e reconhecimento de placas veiculares em tempo real, integrando-o à plataforma *InterSCity* e contribuindo para o aprimoramento de processos de segurança pública, especialmente no combate ao roubo e furto de veículos.

Objetivos Específicos

- Desenvolver um pipeline completo de aquisição e processamento de vídeo, abrangendo detecção de veículos, extração de regiões de interesse e reconhecimento automático de placas;
- Implementar mecanismos capazes de identificar automaticamente o padrão da placa brasileira (Mercosul ou modelo antigo), aplicando máscaras específicas para correção de ambiguidades no OCR;
- Implementar uma abordagem baseada em Sensoriamento Móvel Participativo para coleta de vídeos georreferenciados em Cidades Inteligentes;
- Realizar integração com a plataforma *InterSCity*, permitindo o armazenamento, consulta e correlação dos eventos detectados;
- Avaliar o desempenho do sistema utilizando métricas de desempenho preditivo (Precision, Recall e F1-Score) e métricas de consumo de recursos (CPU, memória, bateria e dados móveis), comparando os cenários de processamento local e remoto em condições reais de tráfego urbano.

1.6 Contribuições

As principais contribuições deste trabalho podem ser sintetizadas da seguinte forma:

- Proposição de um sistema participativo de rastreamento veicular que reduz a dependência de infraestrutura fixa e amplia a cobertura espacial do monitoramento urbano;
- Implementação de uma estratégia de reconhecimento de placas aprimorada, baseada na identificação automática do padrão de placas brasileiras (modelo antigo e padrão

Mercosul) e no uso de máscaras especializadas para correção de ambiguidades alfanuméricas;

- Desenvolvimento de um protótipo acadêmico funcional, denominado *SmarTTrack*, concebido exclusivamente para fins de pesquisa científica, que integra etapas de detecção e rastreamento de veículos, reconhecimento óptico de caracteres e registro georreferenciado das detecções;
- Integração do sistema com a plataforma *InterSCity*, viabilizando a persistência e recuperação unificada de eventos relevantes por autoridades de segurança pública;
- Avaliação experimental em cenários reais, demonstrando a viabilidade da solução em termos de acurácia, desempenho e operação sob dupla mobilidade.

1.7 Organização do Trabalho

Este trabalho está estruturado da seguinte forma:

- O Capítulo 2 descreve os fundamentos teóricos que embasam este estudo, incluindo visão computacional, *deep learning*, ALPR e o paradigma de Sensoriamento Móvel Participativo;
- O Capítulo 3 apresenta os trabalhos relacionados, discutindo as principais abordagens de detecção, rastreamento e reconhecimento de placas na literatura especializada;
- O Capítulo 4 detalha a metodologia adotada, a arquitetura do sistema *SmarTTrack*, os modelos empregados e os procedimentos de integração com o *InterSCity*;
- O Capítulo 5 apresenta os experimentos realizados e discute os resultados obtidos;
- O Capítulo 6 apresenta as conclusões, limitações observadas e direções para trabalhos futuros.

2 Fundamentação Teórica

Este capítulo apresenta os fundamentos conceituais e tecnológicos que sustentam o desenvolvimento do modelo proposto. São discutidos os conceitos de *Cidades Inteligentes*, sensoriamento participativo e *Mobile Crowdsensing*, princípios de Reconhecimento Automático de Placas Veiculares (ALPR), técnicas modernas de detecção e rastreamento de veículos, mecanismos de Reconhecimento Óptico de Caracteres (OCR), bem como a integração com plataformas digitais de gestão urbana, com ênfase na plataforma *InterSCity*. Tais tópicos fornecem o embasamento necessário para compreender o contexto e a relevância do modelo desenvolvido nesta dissertação.

2.1 Cidades Inteligentes e Mobilidade Urbana

O termo *Cidade Inteligente* é utilizado para representar a ideia de reduzir os problemas decorrentes da urbanização por meio da aplicação de novas tecnologias, tornando o ambiente urbano mais sustentável, eficiente e orientado a dados. Segundo Zheng et al. (ZHENG et al., 2014), as Cidades Inteligentes procuram integrar soluções tecnológicas capazes de melhorar a gestão pública, otimizar serviços e promover qualidade de vida.

Dentre as características centrais de uma Cidade Inteligente, destaca-se o uso pervasivo de Tecnologias da Informação e Comunicação (TIC) em diferentes domínios urbanos para aumentar a eficiência no consumo de recursos, aprimorar processos e apoiar a tomada de decisão. A análise de Neirotti et al. (NEIROTTI et al., 2014) mostra que iniciativas de Cidades Inteligentes distribuem-se por múltiplos domínios — como mobilidade, energia, meio ambiente, governança e aspectos sociais — nos quais as TIC atuam como tecnologias habilitadoras. Os autores também ressaltam a heterogeneidade observada entre as cidades analisadas, explicando que fatores como prioridades locais, restrições e condições institucionais influenciam a forma como cada município desenvolve suas iniciativas, o que afasta a existência de um modelo único de Cidade Inteligente.

A Internet das Coisas (IoT) é reconhecida como uma das tecnologias-chave para a implementação desse paradigma, pois permite a conexão de sensores, dispositivos e infraestruturas capazes de coletar, processar e transmitir dados em tempo real. Essa camada de conectividade amplia a capacidade das cidades de monitorar fenômenos urbanos e sustenta aplicações inteligentes em áreas como mobilidade, monitoramento ambiental, gestão de energia e segurança pública.

Essa visão dialoga diretamente com Hernández-Muñoz et al. (HERNÁNDEZ-MUÑOZ et al., 2011), que afirmam que, em um sentido holístico, as cidades podem

ser compreendidas como *sistemas de sistemas*. Essa perspectiva sistêmica fundamenta a arquitetura das Cidades Inteligentes, nas quais diferentes infraestruturas urbanas — transporte, comunicação, segurança, serviços administrativos e sensores ambientais — passam a operar de maneira integrada por meio de tecnologias associadas à Internet do Futuro. Os autores destacam que a IoT, aliada a plataformas interoperáveis e serviços urbanos inteligentes, constitui a base para a coleta, o processamento e a disseminação contínua de informações em tempo real.

No contexto brasileiro, Kon e Santana (KON; SANTANA, 2016) reforçam que o desenvolvimento de Cidades Inteligentes depende não apenas da adoção de tecnologias, mas de arranjos sociotécnicos que envolvem governança digital, interoperabilidade entre sistemas, transparência e participação social. Para os autores, a fragmentação tecnológica — marcada por sistemas isolados, soluções proprietárias e ausência de padrões — é um dos principais obstáculos à consolidação de cidades verdadeiramente inteligentes. Assim, a cidade só se torna inteligente quando suas tecnologias são orientadas ao interesse público de maneira coordenada e interoperável.

A literatura institucional brasileira corrobora essa visão. A *Carta Brasileira para Cidades Inteligentes* (Brasil, 2020), citada na dissertação de Silva (SILVA, 2024), define que uma Cidade Inteligente é aquela que promove transformação digital sustentável nos eixos econômico, ambiental e sociocultural, articulando inovação, inclusão, governança, acessibilidade e participação cidadã. O documento enfatiza princípios como interoperabilidade, planejamento baseado em dados, integração entre serviços e foco no bem-estar da população.

Do ponto de vista tecnológico, avanços recentes em IoT, visão computacional e aprendizagem de máquina têm expandido a capacidade das cidades de monitorar fenômenos urbanos em larga escala. Zheng et al. (ZHENG et al., 2014) destacam que sistemas inteligentes baseados em sensores conectados e análise preditiva permitem compreender padrões urbanos, otimizar recursos e responder a eventos críticos, como congestionamentos, falhas de infraestrutura e incidentes de segurança pública.

Em síntese, a literatura demonstra que Cidades Inteligentes constituem ecossistemas sociotécnicos baseados na integração entre infraestrutura digital, plataformas interoperáveis, governança orientada a dados e participação cidadã. Nesse contexto, gestores públicos podem se beneficiar do crescimento e aprimoramento dessas infraestruturas para fortalecer políticas de segurança pública. Tecnologias como visão computacional, identificação automática de veículos e sistemas de rastreamento em tempo real ampliam a capacidade de monitoramento urbano e permitem ações preventivas fundamentadas em evidências, contribuindo para a redução de riscos e para a melhoria da qualidade de vida da população.

2.1.1 Elementos Estruturais das Cidades Inteligentes

Com base em Neirotti et al. (NEIROTTI et al., 2014), Hernández-Muñoz et al. (HERNÁNDEZ-MUÑOZ et al., 2011), Kon e Santana (KON; SANTANA, 2016) e nas diretrizes da *Carta Brasileira para Cidades Inteligentes* (Brasil, 2020), é possível identificar elementos estruturais recorrentes no desenvolvimento de Cidades Inteligentes. Embora não exista um modelo único, os trabalhos analisados apontam para cinco dimensões centrais:

1. **Infraestrutura digital e sensoriamento:** redes de comunicação, dispositivos IoT, sensores distribuídos, câmeras, plataformas de coleta e monitoramento;
2. **Interoperabilidade e governança de dados:** padrões de integração, APIs abertas, segurança da informação, transparência e mecanismos de compartilhamento entre serviços públicos;
3. **Inteligência urbana:** uso de algoritmos de aprendizagem de máquina, análise preditiva e visão computacional para transformar dados em conhecimento acionável;
4. **Participação cidadã e inovação social:** plataformas colaborativas, engajamento digital e soluções baseadas em sensoriamento distribuído;
5. **Sustentabilidade e resiliência:** planejamento urbano integrado que abranja mobilidade, saneamento, energia, segurança e adaptação a riscos ambientais.

Esses elementos constituem a base conceitual que permite compreender a inserção de sistemas de monitoramento veicular distribuído — como o *SmarTTrack* — no ecossistema das Cidades Inteligentes, especialmente nos domínios da mobilidade e da segurança pública.

2.2 Sensoriamento Participativo e Sensoriamento Móvel Participativo

O avanço dos dispositivos móveis, especialmente dos *smartphones*, possibilitou uma nova forma de coleta distribuída de informações em ambientes urbanos. As primeiras formulações desse paradigma surgiram sob o termo *participatory sensing* (sensoriamento participativo), conceito introduzido por Burke et al. (BURKE et al., 2006) e aprofundado por Lane et al. (LANE et al., 2010). Nesse modelo inicial, indivíduos utilizavam sensores embarcados em seus dispositivos — como câmeras, microfones ou GPS — para coletar dados de maneira voluntária, enviando-os posteriormente para sistemas centrais de análise. A ênfase estava na participação explícita do usuário e na capacidade de gerar dados contextualizados a partir de experiências cotidianas.

Embora inovador, o sensoriamento participativo apresentava limitações importantes: dependia fortemente da iniciativa humana, operava com baixa automação e não conseguia escalar de forma eficiente para aplicações que exigem coleta contínua, heterogênea e em larga escala. Essas limitações motivaram o surgimento de uma evolução conceitual: o **Sensoriamento Móvel Participativo** (*Mobile Crowdsensing*), formalizado por Ganti, Ye e Lei ([GANTI; YE; LEI, 2011](#)). Nesse modelo ampliado, dispositivos móveis atuam simultaneamente como sensores e unidades computacionais, capazes de coletar, processar e transmitir dados de maneira tanto voluntária quanto oportunística, com mínima intervenção do usuário.

O Sensoriamento Móvel Participativo diferencia-se, portanto, por incorporar mecanismos automatizados de captura, permitir participação transparente do usuário e operar de forma massivamente distribuída, aproveitando a ubiquidade dos dispositivos móveis na sociedade contemporânea. Essa capacidade de combinar participação humana, automação e escala transforma o paradigma em uma tecnologia-chave para aplicações modernas em Cidades Inteligentes, especialmente em áreas que demandam monitoramento contínuo, como mobilidade urbana, segurança pública e detecção de eventos em tempo real.

De acordo com Ganti, Ye e Lei ([GANTI; YE; LEI, 2011](#)), o Sensoriamento Móvel Participativo representa um modelo computacional no qual dispositivos móveis atuam simultaneamente como sensores e nós de processamento, capazes de coletar dados ambientais, contextuais, multimídia e georreferenciados. Essa abordagem combina ubiquidade, escalabilidade e baixo custo, uma vez que aproveita recursos já presentes na infraestrutura cotidiana da sociedade. Lane et al. ([LANE et al., 2010](#)) destacam que esse modelo se diferencia do sensoriamento tradicional por integrar tanto a participação humana quanto mecanismos automatizados de coleta, permitindo que sistemas funcionem com diferentes graus de intervenção do usuário.

Características e Arquitetura do Sensoriamento Móvel Participativo

O Sensoriamento Móvel Participativo é tipicamente estruturado em três componentes principais:

- **Sensores móveis:** câmeras, GPS, acelerômetros, giroscópios, microfones e sensores embarcados em veículos;
- **Participantes:** cidadãos, motoristas, ciclistas, agentes de campo, entre outros, que atuam como provedores de dados;
- **Infraestrutura de processamento:** servidores, algoritmos e plataformas urbanas que agregam, analisam e disponibilizam informações.

Segundo Christin ([CHRISTIN, 2016](#)), o Sensoriamento Móvel Participativo pode operar segundo dois modelos distintos de participação. No modelo *participativo*, o usuário decide explicitamente quando e como contribuir com dados, como ocorre, por exemplo, em aplicações nas quais o cidadão aciona voluntariamente a captura de imagens, vídeos ou registros de localização. Já no modelo *oportunistico*, a coleta ocorre de forma contínua e automática após uma configuração inicial, sem intervenção direta do usuário a cada contribuição, sendo comum em aplicações que monitoram tráfego, ruído urbano ou mobilidade por meio de sensores do dispositivo executando em segundo plano.

No contexto deste trabalho, adota-se o modelo *participativo*, uma vez que a coleta de dados somente é iniciada após a ação consciente do usuário ao ativar o sistema de captura. A partir desse acionamento, as etapas subsequentes de processamento são realizadas de forma automática, sem exigir interação contínua, o que reduz a sobrecarga do usuário durante a condução do veículo e preserva a conformidade com aspectos legais e de segurança. Essa escolha busca equilibrar controle do usuário, viabilidade prática e preocupações com privacidade, mantendo os princípios do sensoriamento participativo em aplicações de segurança pública.

Desafios Clássicos no Sensoriamento Móvel Participativo

Revisões sistemáticas da área, como a de Khan et al. ([KHAN et al., 2019](#)), apontam quatro grandes desafios recorrentes:

- **Gestão de tarefas:** distribuição adequada das contribuições para evitar redundância ou lacunas de cobertura;
- **Privacidade e anonimização:** proteção de dados sensíveis, especialmente quando envolvem imagens, trajetórias ou placas veiculares;
- **Confiabilidade dos dados:** validação das contribuições e detecção de erros, ruídos ou comportamentos maliciosos;
- **Incentivos:** motivação à participação contínua por meio de gamificação, recompensas sociais, financeiras ou engajamento comunitário.

Esses desafios tornam-se ainda mais críticos quando o sensoriamento móvel participativo é aplicado a cenários de segurança pública, nos quais dados sensíveis devem ser tratados com rigor técnico e ético.

Rastreamento Móvel Participativo na Mobilidade Urbana e Segurança Pública

A mobilidade urbana é um dos domínios em que o Sensoriamento Móvel Participativo alcançou maior relevância. Aplicações amplamente utilizadas, como *Waze*,

Google Maps e *HereWeGo*, integram dados coletados por milhões de dispositivos para gerar informações atualizadas sobre tráfego, congestionamentos, acidentes e obras. Essa mesma lógica pode ser aplicada ao monitoramento veicular, permitindo localizar e rastrear veículos suspeitos por meio da colaboração de dispositivos distribuídos.

Chen et al. (CHEN et al., 2019) ilustram esse potencial com o sistema *CrowdTracking*, que utiliza fotografias capturadas por cidadãos para localizar veículos em áreas sem cobertura de câmeras fixas. O trabalho demonstra que a capilaridade proporcionada por sensores móveis supera significativamente a de sistemas baseados exclusivamente em infraestrutura estática. Além disso, os resultados mostram que mesmo informações parciais — como a posição aproximada do fotógrafo e o ângulo de captura — podem ser suficientes para estimar a localização de um veículo com erro médio inferior a 6 metros.

Estudos relevantes, como os de Kantarci e Mouftah (KANTARCI; MOUFTAH, 2014) e Jiang et al. (JIANG et al., 2023), reforçam que o Sensoriamento Móvel Participativo pode desempenhar papel essencial em aplicações de segurança pública: análise de rotas suspeitas, monitoramento de veículos suspeitos, identificação de áreas de risco e apoio a operações policiais. A principal vantagem está na descentralização da coleta de dados, que aumenta a cobertura do monitoramento em regiões onde câmeras tradicionais são escassas, defeituosas ou vulneráveis a vandalismo.

Integração com Plataformas Urbanas

Plataformas como o *InterSCity* ampliam o impacto do Sensoriamento Móvel Participativo ao fornecer mecanismos padronizados para registro, agregação e compartilhamento de dados. Em vez de as informações ficarem restritas aos dispositivos individuais, passam a compor um ecossistema urbano interoperável capaz de correlacionar dados provenientes de múltiplas fontes.

Nesse cenário, o sistema *SmarTTrack* utiliza sensoriamento móvel participativo para captar vídeos e eventos veiculares georreferenciados e integrá-los ao *InterSCity*, onde podem ser consultados por agentes públicos. Essa integração possibilita:

- Monitoramento descentralizado e em larga escala;
- Correlação entre sensores móveis e câmeras fixas;
- Organização dos dados por meio de APIs padronizadas;
- Apoio à tomada de decisões em tempo quase real.

A arquitetura distribuída do sensoriamento móvel participativo, somada à robustez da plataforma urbana, cria um ecossistema capaz de apoiar ações de segurança pública

e operações de recuperação de veículos, proporcionando um modelo escalável, de baixo custo e de alto impacto social.

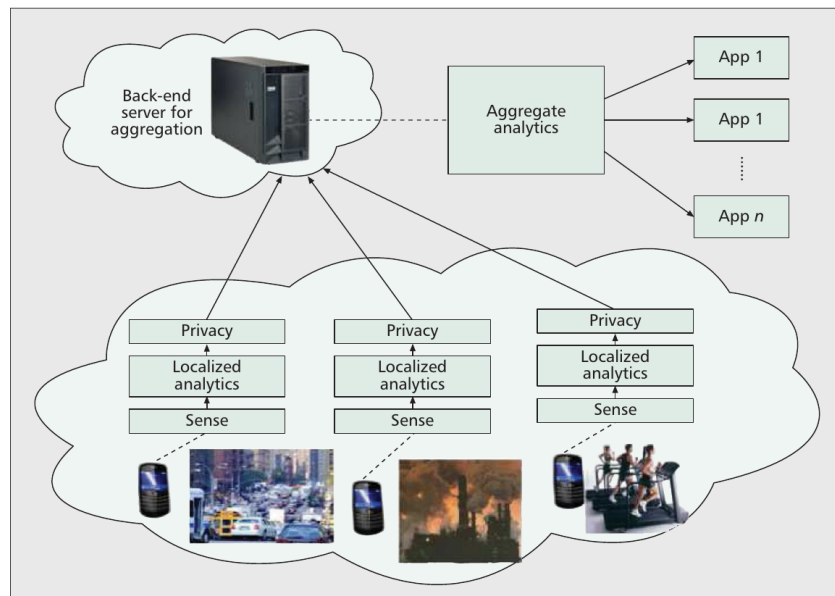


Figura 1 – Fluxo geral de uma aplicação de Sensoriamento Móvel Participativo, no qual dados sensoriais são coletados por dispositivos móveis, processados localmente e enviados ao *back-end* para agregação e mineração. Fonte: (GANTI; YE; LEI, 2011).

2.3 Detecção de Objetos e o Algoritmo YOLO

A detecção automática de objetos é um dos pilares das soluções modernas de visão computacional, permitindo identificar e localizar instâncias de interesse — como veículos, pedestres ou placas veiculares — em imagens ou fluxos de vídeo. Entre os métodos contemporâneos, a família de algoritmos *You Only Look Once* (YOLO), introduzida por Redmon et al. (REDMON et al., 2016), consolidou-se como uma das abordagens amplamente utilizada no meio científico devido à sua capacidade de combinar alto desempenho com operação em tempo real, tornando-se referência no estado da arte em detecção de objetos.

O YOLO representou uma mudança de paradigma ao propor uma arquitetura unificada que trata a detecção como um problema de regressão direta, substituindo abordagens tradicionais baseadas em propostas de região (*region proposals*), como *R-CNN*, *Fast R-CNN* e *Faster R-CNN*. Em vez de executar múltiplas etapas para localizar e classificar regiões de interesse, o YOLO processa a imagem inteira em uma única passagem pela rede neural convolucional, produzindo simultaneamente as caixas delimitadoras (*bounding boxes*) e as respectivas probabilidades de classe. Essa unificação

reduz drasticamente o custo computacional e permite alcançar dezenas ou centenas de quadros por segundo, característica essencial para aplicações móveis e sistemas embarcados.

A arquitetura original divide a imagem em uma grade $S \times S$ (Figura 2), na qual cada célula é responsável por prever caixas delimitadoras, valores de confiança e probabilidades de classe. Essa estratégia mantém coerência espacial e reduz redundâncias comuns em técnicas como janelas deslizantes (*sliding windows*). O modelo proposto por Redmon et al. utilizava 24 camadas convolucionais e duas totalmente conectadas, apresentando desempenho competitivo em bases como PASCAL VOC¹ e COCO².

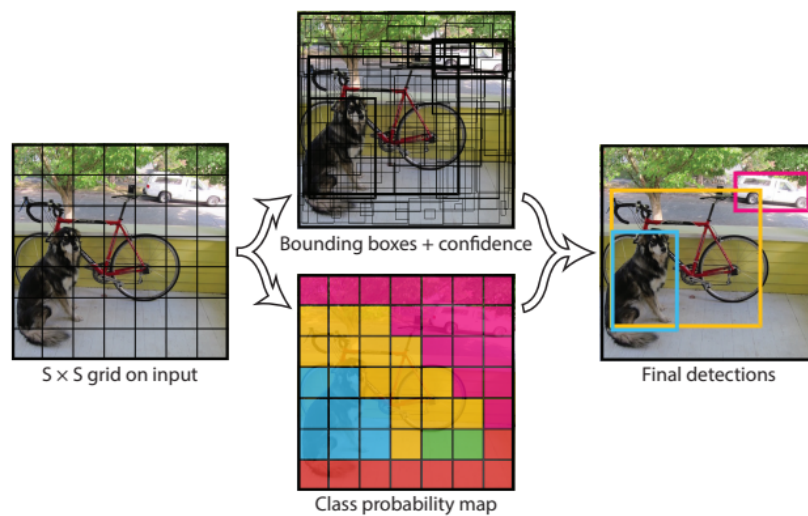


Figura 2 – Funcionamento do algoritmo *YOLO*. Fonte: (REDMON et al., 2016)

Com o tempo, sucessivas versões aperfeiçoaram a arquitetura do YOLO, incorporando novas funções de perda, mecanismos de atenção, *backbones* mais eficientes (como *Darknet* e CSPNet) e melhorias substanciais no tratamento de objetos pequenos — o que inclui placas veiculares — além da integração de funcionalidades voltadas ao rastreamento de objetos em vídeo. A versão mais recente, **YOLOv11**, apresenta avanços estruturais que aumentam a precisão sem comprometer a velocidade, tornando-se especialmente adequada para aplicações em dispositivos móveis e ambientes com restrição de hardware (RAHIMA; HUSSAIN, 2024).

A capacidade de rastreamento nativo foi introduzida oficialmente pela *Ultralytics* a partir do *YOLOv8*. Nessa versão, o algoritmo passou a oferecer um modo específico de rastreamento que associa detecções ao longo do tempo, atribuindo identificadores únicos aos objetos em movimento e permitindo a integração direta entre detecção e acompanhamento em tarefas de múltiplos alvos³. Esse recurso ampliou significativamente o potencial de uso

¹ <<https://docs.ultralytics.com/pt/datasets/detect/voc/>>

² <<https://cocodataset.org/>>

³ Documentação oficial da Ultralytics: <<https://docs.ultralytics.com/modes/track/>>

do YOLO em aplicações de vídeo, especialmente naquelas que demandam análise temporal e continuidade entre quadros.

No contexto do sistema *SmarTTrack*, o algoritmo YOLO é empregado para a detecção de placas veiculares. Um modelo YOLOv11 foi treinado para identificar apenas regiões de placa nas imagens capturadas, fornecendo coordenadas precisas para posterior extração e reconhecimento de caracteres. Essa escolha decorre da capacidade do YOLO de detectar objetos pequenos com elevada robustez, mesmo em cenas dinâmicas nas quais tanto o veículo quanto o dispositivo móvel encontram-se em movimento.

Três características fundamentam a adoção do YOLO no *SmarTTrack*:

1. **Velocidade:** mesmo em modelos compactos, atinge taxas elevadas de quadros por segundo, essenciais para cenários com mobilidade.
2. **Robustez:** mantém desempenho consistente sob variações de iluminação, oclusão parcial e movimentação da câmera.
3. **Aptidão para objetos pequenos:** versões recentes aprimoraram a sensibilidade para detectar elementos como placas veiculares em cenas complexas.

Em síntese, o *YOLO* permanece como uma das abordagens mais consolidadas e eficazes para detecção de objetos em tempo real. Sua evolução contínua, aliada à capacidade de operar em dispositivos de baixo consumo energético, torna-o especialmente relevante para aplicações de sensoriamento participativo situadas no contexto de Cidades Inteligentes, como o *SmarTTrack*, que depende de detecções rápidas, precisas e escaláveis para apoiar ações de segurança pública e recuperação veicular.

2.4 Reconhecimento Automático de Placas Veiculares (ALPR)

O Reconhecimento Automático de Placas Veiculares (ALPR – *Automatic License Plate Recognition*) é amplamente utilizado em sistemas de monitoramento urbano, segurança pública e gestão da mobilidade. O ALPR combina técnicas de visão computacional e aprendizado profundo com o objetivo de localizar a placa, segmentar seus caracteres e reconhecer o conteúdo textual de forma automática. Embora seja estudado há mais de duas décadas, o campo passou por transformações relevantes com a popularização das redes neurais convolucionais e, mais recentemente, dos modelos *fim-a-fim*.

Segundo um survey recente ([SHASHIRANGANA et al., 2021](#)), um sistema ALPR pode ser dividido em quatro etapas principais: (1) captura ou aquisição da imagem, (2) detecção da região da placa, (3) segmentação dos caracteres e (4) reconhecimento (OCR). Apesar de cada etapa ter sido historicamente tratada de forma independente, a literatura

caminha para abordagens unificadas baseadas em aprendizado profundo, nas quais todo o processo é modelado de forma integrada.

Aquisição e Desafios de Captura

A robustez de um sistema ALPR depende, em grande medida, da qualidade das imagens de entrada. Em cenários reais, especialmente em vias urbanas, as condições de captura são desafiadoras: variações bruscas de iluminação (HDR), chuva, trepidações, reflexos metálicos, distância excessiva ou ângulos oblíquos (ZHU et al., 2019). Além disso, sistemas baseados em sensoriamento participativo – como no contexto deste trabalho – envolvem câmeras móveis, o que acrescenta movimento simultâneo da cena e do sensor (dupla mobilidade).

Essas condições impõem desafios que motivaram a adoção de detectores mais robustos, como as variantes modernas de *YOLO*, que apresentam desempenho significativamente superior a técnicas clássicas baseadas em segmentação ou operadores morfológicos.

Deteção da Placa

A deteção da região de interesse (ROI) onde a placa se encontra é, tradicionalmente, uma das etapas mais sensíveis do *pipeline* ALPR. Abordagens clássicas incluem: (1) Transformações morfológicas para localizar regiões retangulares; (2) Operadores de borda (*Sobel*, *Canny*) para enfatizar regiões de alta frequência e (3) Modelos baseados em textura ou histogramas.

Esses métodos, entretanto, apresentam limitações em condições de baixa qualidade ou em cenas complexas. Com o avanço do aprendizado profundo, detectores de objetos como *YOLO*, *SSD* e *Faster R-CNN* tornaram-se a norma, oferecendo maior robustez, melhor generalização e capacidade de processar vídeo em tempo real. Estudos recentes, como os de Kushwaha e Suseela (KUSHWAHA; SUSEELA, 2022), demonstram mAP superior a 99% na deteção de placas utilizando variantes modernas do *YOLO*.

No contexto desta dissertação, emprega-se um modelo específico treinado exclusivamente para a deteção de placas brasileiras – atendendo tanto ao padrão antigo quanto ao padrão Mercosul – o que maximiza a precisão em cenários de tráfego real.

Segmentação de Caracteres

Uma vez localizada a placa, torna-se necessário separar visualmente seus caracteres individuais. Abordagens clássicas tratam a segmentação como um problema de processamento de imagens, utilizando limiarização adaptativa, análise de componentes conectados e operadores morfológicos (ANANTHI et al., 2023).

Em placas veiculares, a segmentação dos caracteres apresenta desafios específicos decorrentes de diferentes fatores observados em cenários reais. Há, inicialmente, a variação de tipografia entre diferentes países e, no caso brasileiro, entre os dois padrões atualmente em circulação: o modelo convencional, que vem sendo descontinuado, e o padrão Mercosul. No padrão Mercosul, a presença da faixa azul na parte superior introduz contrastes adicionais que podem interferir nos métodos de limiarização e na separação dos caracteres. Soma-se a isso a ocorrência de desgaste natural, sujeira, danificações e obstruções parciais, fatores que alteram significativamente o contorno e a legibilidade dos símbolos. Além disso, no padrão antigo, a existência do hífen (ABC-1234), ausente no modelo atual, adiciona uma complexidade extra ao processo de segmentação, exigindo mecanismos capazes de distinguir sinais gráficos de caracteres válidos durante o reconhecimento.

Com a chegada do aprendizado profundo, a segmentação manual vem sendo gradualmente substituída por modelos fim-a-fim que realizam reconhecimento direto da sequência de caracteres, eliminando a etapa intermediária. Essa abordagem já se encontra difundida em técnicas OCR modernas, como o PaddleOCR utilizado neste trabalho.

Reconhecimento dos Caracteres

O reconhecimento textual pode ser realizado por métodos tradicionais – como *Tesseract* ou classificadores baseados em SVM e k-NN – mas esses métodos sofrem diante de variações de rotação, iluminação ou baixa qualidade da imagem.

Modelos modernos, como CNNs combinadas com LSTMs, mecanismos de atenção e CTC (*Connectionist Temporal Classification*), apresentam maior resiliência, especialmente em dados ruidosos ou com sombras. Zhu et al. (ZHU et al., 2019), por exemplo, demonstram a eficácia de um modelo CNN fim-a-fim capaz de reconhecer uma placa completa mesmo em vídeos de baixa qualidade, utilizando integração de resultados ao longo dos quadros.

Integração com Rastreamento e Aplicações Urbanas

A integração entre ALPR e algoritmos de rastreamento (*Multi-Object Tracking*) potencializa a estabilidade do reconhecimento ao longo de vídeos, agregando múltiplas detecções para produzir uma leitura final mais robusta. Zhu et al. (ZHU et al., 2019) demonstram que a fusão de previsões entre quadros pode elevar substancialmente o desempenho do OCR.

Essa integração é fundamental em aplicações de segurança pública, especialmente em cenários de dupla mobilidade, nos quais cada quadro traz pequenas variações de nitidez ou ângulo. No caso desta pesquisa, o ALPR serve como componente central para a identificação de veículos de interesse, permitindo armazenar, no *InterSCity*, informações relevantes para operações policiais, investigações ou análises de mobilidade urbana.

2.5 Rastreamento de Múltiplos Objetos (MOT)

O rastreamento de múltiplos objetos (MOT – *Multi-Object Tracking*) é um componente fundamental em sistemas de monitoramento urbano, especialmente quando o objetivo é acompanhar veículos ao longo do tempo, integrando detecção, trajetória e identificação. Em cenários urbanos reais, o MOT enfrenta desafios significativos decorrentes da alta densidade de tráfego, da oclusão parcial ou total dos objetos, das mudanças abruptas de iluminação e da presença simultânea de múltiplos veículos com dimensões e velocidades distintas.

Tradicionalmente, o problema de MOT é formulado como o processo de estimar, para cada objeto detectado, uma trajetória coerente no espaço-tempo, associando as detecções de cada quadro aos objetos previamente identificados. Conforme apresentado em análises abrangentes da literatura (LUO et al., 2021), o *pipeline* clássico envolve três etapas: (1) detecção de objetos em cada quadro, (2) associação entre detecções e trilhas existentes e (3) atualização das trajetórias. A qualidade da detecção é frequentemente o fator que mais influencia a precisão do rastreamento, especialmente quando os objetos estão próximos entre si ou quando as condições de visibilidade são desfavoráveis.

Métodos Tradicionais de Rastreamento

Abordagens clássicas de rastreamento incluem modelos probabilísticos, como o Filtro de Kalman (SILVA, 2007), que estimam a posição futura de um objeto a partir de seu estado anterior. O Filtro de Kalman fornece uma solução recursiva para problemas de estimação em sistemas dinâmicos lineares sujeitos a ruídos gaussianos, oferecendo previsões eficientes mesmo quando as medições estão incompletas ou degradadas por ruído. Em síntese, o algoritmo combina previsões baseadas em um modelo de movimento com novas observações sensoriais, ponderando ambas de acordo com suas incertezas estimadas (ALMENDRA, 2015; SANTANA, 2011).

No contexto de rastreamento veicular, essa característica torna o Filtro de Kalman especialmente útil para suavizar trajetórias, reduzir oscilações causadas por detecções instáveis e estimar posições mesmo quando um objeto deixa temporariamente de ser detectado em alguns quadros consecutivos. Em sistemas embarcados — incluindo dispositivos móveis — o algoritmo apresenta vantagens adicionais: possui baixo custo computacional, opera de forma *online* e requer pouca memória, sendo adequado para cenários sem disponibilidade de GPUs ou onde arquiteturas modernas de aprendizado profundo não podem ser executadas em tempo real.

Embora técnicas modernas de rastreamento, como o *Deep SORT* ou os módulos nativos de rastreamento do *YOLOv11*, integrem redes neurais profundas para associação de detecções, tais abordagens geralmente exigem maior poder computacional e apresentam

limitada viabilidade em dispositivos móveis. Em ambientes com restrições de hardware — como no caso de aplicações de Sensoriamento Participativo Móvel — a execução contínua desses módulos tende a resultar em queda de desempenho, aumento do consumo energético e impacto direto no tempo de resposta da aplicação. A literatura evidencia que métodos clássicos baseados em modelos de estado continuam desempenhando papel relevante em sistemas móveis e distribuídos (SILVA, 2007; CHEN et al., 2019).

Além disso, técnicas tradicionais como a subtração de fundo⁴ também foram amplamente utilizadas por décadas, sobretudo em câmeras fixas. Contudo, sua eficácia diminui drasticamente quando há movimento da câmera ou mudanças abruptas no cenário (OHEKA; TU, 2018). No rastreamento veicular baseado em SMP, tanto o veículo alvo quanto o dispositivo coletor estão em movimento, tornando o fundo instável e inviabilizando métodos baseados em segmentação estática. Esse cenário reforça a necessidade de metodologias mais robustas, capazes de lidar com a complexidade dinâmica das vias urbanas.

Em síntese, embora modelos modernos de rastreamento utilizem arquiteturas profundas, abordagens clássicas como o Filtro de Kalman continuam fundamentais em aplicações móveis e distribuídas, oferecendo uma combinação de eficiência, baixo consumo computacional e estabilidade — características essenciais para sistemas executados diretamente em smartphones e integrados a ecossistemas de Cidades Inteligentes.

Rastreamento Baseado em Detecção (*Tracking-by-Detection*)

Com o advento do aprendizado profundo, consolidou-se o paradigma *tracking-by-detection*, no qual o rastreamento é executado a partir de detecções feitas quadro a quadro por modelos modernos, como *YOLO*, *SSD* e *Faster R-CNN*. Nesses modelos, a função do rastreador é associar, de forma consistente, as caixas detectadas ao longo do vídeo, formando trilhas que representem trajetórias.

A associação é geralmente modelada como um problema de otimização, resolvido por métodos como:

- **Algoritmo Húngaro** – utilizado para resolver problemas de correspondência entre detecções e trilhas, minimizando uma função de custo (MUNKRES, 1957);
- ***IoU Matching*** – estratégia baseada na interseção sobre união entre caixas (ZHU et al., 2019; BEWLEY et al., 2016);

⁴ Subtração de fundo (*background subtraction*) é uma família de técnicas de visão computacional em que se constrói (ou atualiza) um modelo do *background* e, a cada novo quadro, as regiões/pixels que diferem significativamente desse modelo são classificados como *foreground* (objetos em movimento). Por depender de um fundo aproximadamente estacionário, tende a apresentar melhor desempenho em cenários com câmera fixa, sendo sensível a movimento da câmera e a mudanças abruptas no cenário (OHEKA; TU, 2018; PICCARDI, 2004).

- **Modelos probabilísticos refinados** – incorporam previsões temporais e incertezas (ex.: Filtro de Kalman e técnicas de associação probabilística) (KALMAN, 1960; BAR-SHALOM; FORTMANN, 1988).

Leitores modernos como *DeepSORT*⁵ integram um módulo de reconhecimento visual embutido para distinguir objetos semelhantes, reduzindo casos de troca de identidade. Já modelos recentes como *ByteTrack*⁶ realizam associação mesmo para detecções de baixa confiança, aumentando a estabilidade do rastreamento em ambientes ruidosos.

MOT em Cenários Urbanos e ALPR

Em aplicações urbanas, o rastreamento de veículos deve lidar com múltiplas características simultâneas: veículos que aceleram ou desaceleram bruscamente, motos que se deslocam lateralmente entre faixas, oclusões causadas por ônibus ou caminhões e variações constantes de iluminação. Além disso, a necessidade de sincronização entre rastreamento e reconhecimento de placas adiciona uma camada adicional de complexidade.

Estudos como os de Zhu et al. (ZHU et al., 2019), bem como trabalhos voltados à integração entre rastreamento e reconhecimento em ALPR, indicam que a associação temporal entre quadros contribui para leituras mais estáveis, permitindo agregar múltiplas evidências do mesmo veículo ao longo do tempo e reduzir flutuações entre quadros. Da mesma forma, o trabalho de Chen et al. (CHEN et al., 2019) demonstra a importância do rastreamento para estimar trajetórias espaciais de veículos, ainda que baseado em fotografias estáticas enviadas por cidadãos — um cenário distinto dos sistemas de detecção e reconhecimento contínuos em vídeo.

Sistemas como os de Kiew et al. (KIEW et al., 2021) utilizam apenas *template matching*, uma técnica clássica de detecção baseada na comparação direta entre regiões da imagem e modelos pré-definidos (*templates*). Em essência, o algoritmo percorre o quadro avaliando o grau de similaridade entre cada janela e o padrão esperado, geralmente por correlação ou diferença absoluta de intensidade. Embora simples e eficiente para cenários altamente controlados — como rotas fixas, câmeras estáticas e objetos com pouca variação visual — essa abordagem é extremamente sensível a mudanças de escala, perspectiva, iluminação e oclusões. Consequentemente, seu desempenho degrada rapidamente em

⁵ O *DeepSORT* foi apresentado por Wojke, Bewley e Paulus (2017), que estenderam o algoritmo SORT ao incorporar um módulo de reconhecimento visual baseado em aprendizado profundo para reduzir trocas de identidade. Artigo: W. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” 2017. Disponível em: <<https://arxiv.org/abs/1703.07402>>.

⁶ O *ByteTrack* foi proposto por Y. Xingyi et al. (2022), introduzindo uma estratégia de associação que utiliza detecções de alta e baixa confiança para melhorar a estabilidade do rastreamento em cenários ruidosos. Artigo: Y. Xingyi, Z. Chen, S. Wang, et al., “ByteTrack: Multi-Object Tracking by Associating Every Detection Box,” 2022. Disponível em: <<https://arxiv.org/abs/2110.06864>>.

ambientes urbanos reais, onde veículos apresentam aparências distintas, movimentação irregular e múltiplas fontes de ruído visual.

Por sua vez, abordagens como a de Oheka e Tu ([OHEKA; TU, 2018](#)) empregam detectores baseados em características *Haar* combinadas ao algoritmo *AdaBoost*, formando as chamadas *Haar Cascades*. As características *Haar* capturam contrastes retangulares simples (por exemplo, diferença entre regiões claras e escuras), enquanto o *AdaBoost* seleciona e pondera automaticamente os classificadores fracos mais discriminativos, construindo um classificador mais robusto. Embora essa técnica tenha sido importante historicamente — especialmente na detecção de faces — ela depende de padrões visuais muito simples e não se adapta bem às variações complexas presentes no tráfego real, como veículos de diferentes formatos, ângulos de visão diversificados, reflexos, sombras e sobreposições. Dessa forma, ainda que leve e computacionalmente eficiente, a combinação *Haar* e *AdaBoost* não apresenta robustez suficiente para cenas com múltiplos veículos em movimento ou ambientes dinâmicos típicos de vídeos embarcados.

Rastreamento com Câmera Móvel (Dupla Mobilidade)

O rastreamento com câmera fixa já é desafiador; com câmera móvel, o problema torna-se muito mais complexo. Em situações de sensoriamento participativo, como as consideradas nesta dissertação, o dispositivo não apenas observa veículos em movimento, mas também se desloca simultaneamente, criando um cenário de dupla mobilidade. Isso gera:

- Variação contínua do fundo da cena;
- Mudança constante na perspectiva, com aproximações e afastamentos rápidos;
- Flutuações de luminosidade devido à posição da câmera;
- Oclusões dinâmicas, sobretudo em vias congestionadas.

Poucos trabalhos abordam explicitamente esse cenário, que é central para sistemas baseados em SMP. Este aspecto motiva o desenvolvimento de *pipelines* mais robustos, como o adotado no *SmarTTrack*, em que o YOLOv11 é empregado para garantir detecções estáveis e contínuas ao longo de todo o vídeo, mesmo sob movimento intenso.

Integração do MOT com ALPR e Plataformas Urbanas

Ao integrar MOT com ALPR, o sistema passa a extrair não apenas a trajetória espacial dos veículos, mas também uma identidade semântica associada a cada deslocamento — a placa. Essa vinculação enriquece significativamente o valor informacional do

rastreamento, permitindo reconstruir rotas de forma precisa, gerar registros estruturados do comportamento veicular e apoiar ações de segurança pública e gestão de tráfego. Informações desse tipo são particularmente relevantes em contextos operacionais, pois fornecem insumos objetivos para análises estratégicas, investigação de ocorrências e compreensão de dinâmicas urbanas.

Além disso, quando combinado a plataformas urbanas como o *InterSCity*, o rastreamento multiveicular deixa de ser um módulo isolado e passa a integrar um ecossistema maior de serviços urbanos. Os eventos gerados — trajetórias, placas reconhecidas e metadados contextuais — podem ser persistidos de forma padronizada, interoperável e acessível a agentes públicos, fortalecendo iniciativas de monitoramento, tomada de decisão e políticas de mobilidade. Assim, a integração entre MOT, ALPR e plataformas urbanas contribui diretamente para a construção de arquiteturas inteligentes de gestão urbana, alinhadas com os princípios das Cidades Inteligentes.

2.6 Reconhecimento Óptico de Caracteres

O Reconhecimento Óptico de Caracteres (OCR – *Optical Character Recognition*) constitui uma etapa essencial em sistemas de ALPR, pois transforma a região da placa veicular em texto estruturado. Técnicas clássicas baseadas em limiarização e segmentação apresentam limitações significativas em cenários reais, especialmente diante de ruído, baixa resolução, distorções geométricas e variações de iluminação. Por essa razão, arquiteturas recentes apoiadas em aprendizagem profunda têm se destacado por oferecer maior robustez e desempenho em ambientes urbanos dinâmicos. Como demonstrado por Jørgensen ([JØRGENSEN, 2017](#)), abordagens tradicionais sofrem degradação acentuada em condições adversas, ao passo que métodos baseados em aprendizado profundo mantêm resultados mais estáveis. Em aplicações com câmeras móveis — como viaturas ou dispositivos embarcados — essa resiliência torna-se indispensável para assegurar precisão e consistência no reconhecimento das placas.

Evolução do OCR e Limitações de Abordagens Tradicionais

Métodos convencionais como *Tesseract*, SVMs e K-NN foram amplamente utilizados na literatura, mas apresentam desempenho limitado em ambientes não controlados. Em placas veiculares, onde sombras, reflexos, sujeira e vibrações são frequentes, tais métodos falham em reconhecer caracteres com confiabilidade adequada ([SHASHIRANGANA et al., 2021](#); [JØRGENSEN, 2017](#); [Laroca et al., 2018](#)). Além disso, abordagens tradicionais dependem fortemente de uma segmentação perfeita caractere a caractere – uma etapa vulnerável a erros acumulados.

Com o avanço das redes neurais convolucionais e recorrentes, modelos modernos

passaram a realizar a leitura de texto em um fluxo *fim-a-fim*, combinando extração de características, normalização espacial e decodificação sequencial sem necessidade de segmentação explícita. Essa abordagem tornou o OCR mais robusto, especialmente em textos curtos e estruturados, como placas veiculares.

O Sistema PP-OCR

O *PaddleOCR*, desenvolvido pela equipe *PaddlePaddle* da Baidu⁷, representa um dos sistemas mais eficientes e amplamente utilizados para OCR industrial. O modelo PP-OCR, detalhado por Du et al. (DU et al., 2020), foi projetado para aplicações reais com restrições de memória e tempo de inferência, combinando alta precisão, leveza e suporte a múltiplos alfabetos.

A arquitetura PP-OCR é composta por três módulos principais:

1. **Detecção de Texto (PP-OCR-Det)** – baseada em um detector derivado do DB (*Differentiable Binarization*), que produz mapas de probabilidade refinados capazes de localizar regiões textuais mesmo em ambientes complexos;
2. **Reconhecimento de Texto (PP-OCR-Rec)** – que combina uma CNN para extração de características, seguida de um codificador bidirecional e um decodificador baseado em CTC (*Connectionist Temporal Classification*);
3. **Mecanismos de correção e pós-processamento** – incluindo modelos de linguagem leves e ajustes heurísticos para refinar resultados.

O conjunto oferece uma solução robusta para textos curtos e estruturados, com desempenho eficiente em hardware limitado, fator que o torna adequado para dispositivos móveis e *pipelines* de sensoriamento participativo.

Aplicação do PaddleOCR em Placas Veiculares

Embora o *PaddleOCR* seja originalmente um OCR genérico, sua arquitetura modular permite adaptações eficientes para domínios especializados, como o reconhecimento de placas veiculares. O módulo de reconhecimento do PP-OCR (*PP-OCR-Rec*) segue exatamente a combinação de CNN, LSTM bidirecional e decodificação por CTC descrita por Du et al. (DU et al., 2020). Essa mesma arquitetura tem sido destacada em diversos estudos de ALPR (CALITZ; HILL, 2020; PUARUNGROJ; BOONSIRISUMPUN, 2018) por sua robustez em cenários reais, incluindo variação tipográfica, borrões causados por movimento, reflexos, deformações e ambiguidades visuais entre caracteres.

⁷ Baidu é uma empresa chinesa de tecnologia, desenvolvedora da plataforma *PaddlePaddle*. Informações disponíveis em: <<https://www.baidu.com>> e <<https://www.paddlepaddle.org.cn>>.

Além disso, pesquisas recentes em ALPR, como as de Ananthi et al. (ANANTHI et al., 2023), Calitz e Hill (CALITZ; HILL, 2020) e Zhu et al. (ZHU et al., 2019), mostram que essas arquiteturas conseguem lidar com reflexos, pequenas deformações e ambiguidades recorrentes entre caracteres visualmente semelhantes (O/0, 1/I, 5/S e B/8).

No contexto do PP-OCR, o módulo de detecção baseado em DB (*Differentiable Binarization*) contribui para isolar com maior precisão a região da placa, mesmo em ambientes ruidosos. Em seguida, o módulo de reconhecimento processa toda a sequência de caracteres de uma só vez, sem exigir etapas intermediárias de segmentação. Isso evita problemas comuns em técnicas clássicas de OCR, que dependem da separação exata de cada caractere e, por isso, sofrem em situações de movimento, baixa resolução ou tipografia inconsistente.

Máscaras Semânticas para Placas Brasileiras

O Brasil possui dois padrões oficiais de placas veiculares: o modelo antigo (formato LLL-NNNN) e o padrão Mercosul (formato LLLNLNN). A coexistência desses modelos cria problemas específicos de OCR, uma vez que ambiguidades entre caracteres podem resultar em combinações inválidas. Para mitigar essas limitações, este trabalho emprega **máscaras semânticas** que mapeiam posições permitidas para letras e números, validando e corrigindo automaticamente resultados inconsistentes.

Essa estratégia, recomendada na literatura (CHEN et al., 2021; SHASHIRANGANA et al., 2021), aumenta a confiabilidade do reconhecimento final e reduz significativamente falsos positivos.

Vantagens do *PaddleOCR* para Sensoriamento Participativo

No contexto de SMP, o uso do *PaddleOCR* oferece vantagens específicas, que derivam de sua arquitetura projetada para aplicações em dispositivos de borda. Modelos como o PP-OCRv3 combinam *backbones* leves (por exemplo, *MobileNetV3*), detectores eficientes baseados em binarização diferenciável (DBNet) e reconhecedores modulares (CRNN, SAR, SVTR), permitindo execução em tempo real mesmo em *hardware* móvel (DU et al., 2022; LIAO et al., 2019). Além disso, no treinamento desses modelos é comum o emprego de *data augmentation* (aumento artificial do conjunto de treino por transformações como variações de iluminação, rotações e desfoque), o que contribui para maior robustez a ruído, vibração e distorções típicas de vídeos capturados em movimento. A modularidade da arquitetura — com detectores, vocabulários (dicionários) e reconhecedores substituíveis — facilita o ajuste fino para domínios específicos, incluindo padrões de placas veiculares distintos. Essas características fazem do *PaddleOCR* uma opção tecnicamente adequada para cenários de sensoriamento participativo e segurança pública.

2.7 Integração com Plataformas Urbanas: A Arquitetura *InterSCity*

A consolidação das Cidades Inteligentes depende cada vez mais de plataformas urbanas que permitam integrar de forma transparente diferentes fontes de dados, sensores heterogêneos, aplicações distribuídas e serviços voltados ao monitoramento e à gestão de ambientes urbanos. Em vez de soluções isoladas e construídas *ad-hoc*, o cenário contemporâneo demanda infraestruturas padronizadas, escaláveis e capazes de fornecer interoperabilidade entre sistemas. Nesse contexto, a plataforma *InterSCity* surge como uma proposta robusta e amplamente referenciada na literatura, oferecendo uma arquitetura moderna baseada em microserviços, orientada à experimentação científica e ao desenvolvimento de aplicações urbanas inovadoras.

A Plataforma *InterSCity*

A *InterSCity*, proposta por Esposte et al. (ESPOSTE et al., 2017), é uma plataforma aberta e extensível desenvolvida pelo INCT da Internet do Futuro. A plataforma foi concebida para apoiar aplicações distribuídas em domínios como mobilidade, saúde, segurança pública, energia e outros serviços críticos das Cidades Inteligentes. Sua arquitetura segue o padrão orientado a microserviços, no qual cada funcionalidade é encapsulada como um serviço independente, permitindo escalabilidade horizontal e alta flexibilidade para integração de novos módulos.

A plataforma organiza seus serviços em quatro módulos principais:

1. **Resource Catalog Service:** responsável por registrar, consultar e gerenciar metadados sobre os recursos urbanos;
2. **Data Collector Service:** responsável pela recepção de dados enviados pelos dispositivos distribuídos pela cidade;
3. **Data Analysis Service:** destinado ao processamento analítico e execução de algoritmos de inferência;
4. **Actuator Service:** permite emitir comandos para dispositivos conectados.

Todos os serviços são acessados por meio de APIs REST padronizadas, o que facilita a integração com aplicações externas e garante independência tecnológica.

Interoperabilidade e Escalabilidade em Sistemas Urbanos

A interoperabilidade é um dos pilares fundamentais para sistemas de Cidades Inteligentes. Soluções desenvolvidas de forma isolada tendem a gerar fragmentação e baixa eficiência operacional. A *InterSCity* foi projetada justamente para evitar esse problema,

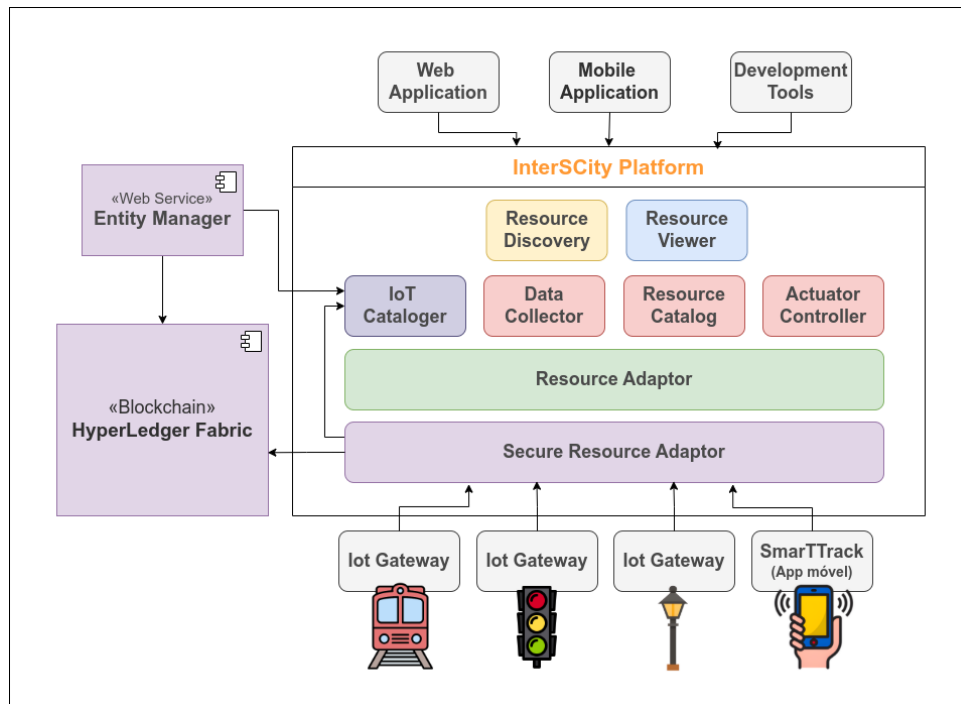


Figura 3 – Fonte: Adaptado de (CARDOSO, 2024)

oferecendo uma camada intermediária que unifica o acesso, armazenamento e gerenciamento de recursos urbanos. Esse modelo propicia: (1) escalabilidade horizontal, (2) padronização de dados, (3) reutilização de serviços e (4) gerenciamento distribuído.

A adoção de microserviços fundamenta a capacidade da plataforma em atender cenários com milhões de eventos, como fluxos contínuos de sensores, câmeras conectadas, veículos inteligentes ou aplicações de *crowdsensing*.

2.8 Considerações Finais

A Fundamentação Teórica apresentada neste capítulo delineou os conceitos, tecnologias e arquiteturas que sustentam o desenvolvimento do modelo *SmarTTrack* e sua inserção no ecossistema das Cidades Inteligentes. Inicialmente, discutiu-se como a integração entre infraestrutura digital, governança orientada a dados, participação cidadã e plataformas interoperáveis constitui elemento central para a consolidação de ambientes urbanos mais eficientes, seguros e responsivos. Nesse contexto, o Sensoriamento Móvel Participativo emerge como uma tecnologia estratégica, capaz de ampliar a capilaridade da coleta de dados e complementar a infraestrutura fixa tradicional, especialmente em domínios como mobilidade e segurança pública.

Foram apresentados, ainda, os fundamentos técnicos associados à detecção de objetos, ao rastreamento de múltiplos objetos e ao Reconhecimento Automático de Placas Veiculares (ALPR), destacando-se os avanços proporcionados por arquiteturas modernas

como o *YOLO* e sistemas OCR baseados em aprendizado profundo, como o *PaddleOCR*. Esses modelos permitem lidar de maneira satisfatória com desafios típicos de cenários urbanos dinâmicos — ruído, variação de iluminação, movimento simultâneo entre sensor e alvo, oclusões e baixa resolução — aspectos essenciais para aplicações baseadas em dispositivos móveis.

Demonstrou-se, ainda, a relevância de plataformas urbanas como a *InterSCity*, projetadas para promover interoperabilidade, escalabilidade e integração de serviços distribuídos. Ao oferecer mecanismos padronizados para registro, agregação e consulta de dados, tais plataformas possibilitam que aplicações de sensoriamento participativo, como o *SmarTTrack*, transcendam soluções isoladas e passem a compor infraestruturas urbanas mais amplas, alinhadas aos princípios das Cidades Inteligentes.

Em síntese, os tópicos abordados estabelecem o arcabouço conceitual e tecnológico necessário para compreender as decisões metodológicas adotadas nesta pesquisa. O próximo capítulo apresenta os trabalhos relacionados, discutindo abordagens e soluções existentes para detecção, rastreamento e reconhecimento de veículos e placas, bem como estratégias de integração em cenários de Cidades Inteligentes, situando o *SmarTTrack* no estado da arte.

3 Trabalhos Relacionados

Este capítulo apresenta uma análise aprofundada dos principais trabalhos relacionados ao rastreamento e reconhecimento de veículos em ambientes urbanos, com ênfase em soluções aplicadas a Cidades Inteligentes. O objetivo é situar o *SmarTTrack* no estado da arte, destacando limitações ainda existentes e possíveis lacunas tecnológicas que motivam a presente pesquisa.

3.1 Metodologia para Seleção dos Trabalhos Relacionados

A seleção dos estudos foi realizada a partir de uma revisão sistematizada da literatura, apoiado pela plataforma *Parsif.al*¹, que permite organizar fluxos de busca, triagem, leitura e extração de informações. A string de busca utilizada foi: “*vehicle tracking*” AND “*camera**”, aplicada em bases científicas internacionais. O período considerado incluiu publicações de 2018 a 2024, quando modelos de aprendizado profundo passaram a consolidar-se como principais soluções para detecção, rastreamento e reconhecimento de placas.

Foram definidos critérios de inclusão que contemplam estudos com foco explícito em rastreamento de veículos a partir de placas veiculares ou que abordassem diretamente o uso de Sensoriamento Móvel Participativo aplicado ao contexto urbano. Dessa forma, priorizaram-se trabalhos que exploram a identificação, o acompanhamento ou a correlação temporal de veículos por meio da leitura de placas, bem como propostas que utilizam dispositivos móveis como sensores distribuídos em ambientes urbanos, com potencial aplicação em Cidades Inteligentes. Como critérios de exclusão, eliminaram-se artigos com menos de seis páginas, estudos que não apresentavam avaliação experimental e publicações em idiomas diferentes de português ou inglês.

A busca inicial retornou 903 artigos; após a remoção automática de duplicidades pelo *Parsif.al*, restaram 600 estudos únicos. A partir dos critérios definidos, 66 artigos foram considerados elegíveis para leitura detalhada. Entre eles, apenas um conjunto reduzido apresentou similaridade técnica direta com a proposta deste trabalho, sendo estes os estudos analisados em detalhe e comparados na Tabela 1. Os trabalhos selecionados são apresentados a seguir, com análise crítica ampliada.

¹ O *Parsif.al* é uma plataforma online de apoio à condução de revisões sistemáticas da literatura, permitindo a definição de protocolos de busca, critérios de inclusão e exclusão, extração de dados e rastreabilidade das decisões ao longo do processo. Disponível em: <<https://parsif.al>>.

3.2 CrowdTracking: Real-Time Vehicle Tracking Through Mobile Crowdsensing

Chen et al. (CHEN et al., 2019) descrevem o *CrowdTracking*, um sistema baseado em sensoriamento participativo projetado para realizar o rastreamento de veículos a partir de imagens capturadas por cidadãos. O sistema utiliza fotografias capturadas por cidadãos, por meio de *smartphones* montados em veículos, como fonte primária de dados para localizar e acompanhar o movimento de veículos-alvo em ambientes urbanos.

O *CrowdTracking* opera a partir de um pipeline composto por quatro etapas principais: (1) coleta de fotografias pelos participantes; (2) estimativa inicial da localização do veículo utilizando sensores embarcados, como GPS, bússola digital e acelerômetro; (3) refinamento da estimativa por meio de múltiplos filtros; e (4) predição da trajetória utilizando alocação dinâmica de rastreadores, priorizando aqueles com maior probabilidade de reencontrar o veículo. Essa estrutura procura produzir estimativas contínuas de movimento mesmo quando as contribuições dos usuários são esparsas.

Para determinar a localização do veículo em cada fotografia, os autores propõem o método de Localização Geométrica Básica (*Basic Geometric Localization* – BGL), que calcula os pontos de interseção entre uma semirreta proveniente da posição do fotógrafo (obtida via GPS) e a malha viária, considerando o ângulo de captura da imagem. Múltiplos pontos de interseção podem surgir, mas apenas um corresponde à posição real do veículo. Para aprimorar essa estimativa, três filtros são aplicados sequencialmente: um filtro baseado em distância, que elimina pontos muito distantes do observador; um filtro topológico, que analisa conectividade e direção dos segmentos viários; e um filtro baseado em tráfego, que considera a velocidade estimada do veículo e os limites de velocidade das vias. O resultado desse processo é uma estimativa refinada denominada *footprint*², que reúne coordenadas geográficas, segmento de estrada, velocidade e direção do veículo.

A avaliação experimental do *CrowdTracking* utiliza dois conjuntos de dados: um real, aplicado ao método de localização, e um dataset sintético criado pelos autores para testar o rastreamento. O *dataset* sintético contém trajetórias de 1.000 táxis em Pequim durante uma semana, usado para treinar os pesos das bordas da rede rodoviária, enquanto outros 100 táxis são tratados como veículos rastreados, oferecendo trajetórias consideradas como verdade de referência. Os *datasets*, contudo, não foram disponibilizados publicamente. Os resultados mostraram que o método de localização alcançou erro médio de aproximadamente 6 metros, e que o sistema de rastreamento obteve desempenho eficiente mesmo com participação limitada dos usuários, reforçando a viabilidade do sensoriamento participativo para ampliar a cobertura de monitoramento urbano.

² Os autores definem um *footprint* como um registro composto pelas coordenadas GPS estimadas, o segmento viário ao qual o veículo pertence, sua direção e velocidade de deslocamento.

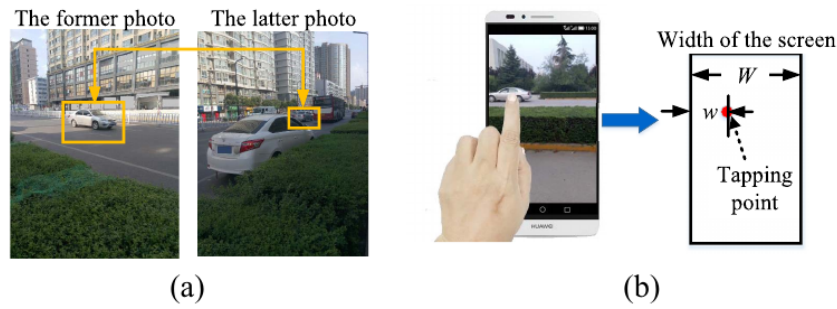


Figura 4 – Ilustração do processo de marcação manual do veículo no sistema *CrowdTracking*, adaptado de Chen et al. (CHEN et al., 2019). (a) O veículo é marcado em duas fotografias capturadas pelos usuários. (b) O usuário toca na tela para selecionar manualmente o veículo. A imagem evidencia a necessidade de interação em tempo real por parte do participante, uma limitação significativa do método.

No *CrowdTracking*, o processo de rastreamento é conduzido a partir de um *pipeline* baseado em participação ativa dos usuários e inferência probabilística sobre a rede viária. Inicialmente, a última observação confirmada do veículo é utilizada para estimar sua posição mais provável e as rotas candidatas que podem ter sido percorridas, considerando a topologia do mapa urbano e pesos aprendidos a partir de trajetórias históricas. Com base nessa estimativa, o sistema seleciona dinamicamente um subconjunto de usuários que se encontram próximos às regiões onde o veículo possui maior probabilidade de passagem. Esses usuários recebem notificações explícitas solicitando a observação visual do veículo e o envio manual de uma fotografia para confirmação do avistamento. Caso uma nova confirmação seja recebida, o modelo atualiza a estimativa de localização e reinicia o ciclo de seleção de usuários; caso contrário, a incerteza é propagada ao longo das rotas possíveis.

Apesar de sua contribuição relevante, o *CrowdTracking* apresenta limitações importantes quando comparado ao escopo deste trabalho. O sistema depende da confirmação manual dos usuários sempre que um veículo é avistado (Figura 4), o que compromete sua escalabilidade e inviabiliza seu uso contínuo, especialmente em países onde o uso de celulares ao dirigir é proibido, como no Brasil. Além disso, o sistema opera com fotografias estáticas, não realizando detecção automática nem reconhecimento de placas, tampouco processando vídeos em fluxo contínuo ou operando sob dupla mobilidade (movimento simultâneo do veículo monitorado e do dispositivo de captura).

Em contraste, o *SmartTrack* automatiza completamente a detecção, rastreamento e identificação de placas, permitindo operação contínua em tempo real e reduzindo dependência do usuário. A abordagem proposta nesta dissertação expande o conceito introduzido pelo *CrowdTracking*, preservando a essência colaborativa do sensoriamento participativo, mas incorporando técnicas de visão computacional baseadas em *YOLO* e OCR, além de integração nativa com plataforma urbana padronizada (*InterSCity*),

ampliando significativamente a aplicabilidade prática da solução em cenários de segurança pública.

3.3 Smart Cities Mobility Monitoring Through Automatic License Plate Recognition and Vehicle Discrimination

Spanu et al. (SPANU et al., 2021) apresentam um sistema destinado ao monitoramento da mobilidade urbana em Cidades Inteligentes, combinando técnicas de Reconhecimento Automático de Placas de Veículos (ALPR) e detecção de objetos com discriminação de veículos baseada na arquitetura *YOLO*. A proposta busca gerar indicadores detalhados sobre o fluxo veicular, densidade de tráfego e padrões de mobilidade, utilizando dados anonimizados em conformidade com regulamentações internacionais de proteção de dados, como a GDPR³. Diferentemente de abordagens orientadas à vigilância ou à segurança pública, o foco do trabalho está na produção de estatísticas e no suporte a sistemas de gestão inteligente do tráfego.

A metodologia adotada pelos autores envolve um pipeline composto por aquisição das imagens, pré-processamento, extração e reconhecimento das placas, discriminação do tipo de veículo e anonimização dos resultados. As imagens são capturadas por câmeras instaladas em pontos estratégicos da cidade, tais como cruzamentos e vias arteriais, sendo processadas em tempo real em ambientes de borda (*edge computing*). Após a detecção e reconhecimento da placa, os dados são anonimizados por meio do algoritmo criptográfico SHA-512⁴. Essa etapa garante que nenhuma informação sensível seja transmitida ou armazenada de forma identificável, atendendo às exigências legais de privacidade em sistemas urbanos.

O sistema foi avaliado utilizando dados coletados em cenários urbanos reais, incluindo vias de tráfego intenso e condições variadas de iluminação. Para aquisição, os autores empregaram câmeras industriais Gifran⁵, enquanto o processamento em tempo real foi realizado por meio do módulo Nvidia Jetson AGX⁶, um hardware amplamente utilizado em aplicações de visão computacional embarcada. Os autores, entretanto, não disponibilizam detalhes completos dos datasets utilizados, como número total de imagens, condições ambientais ou disponibilidade pública, o que limita a reprodutibilidade dos

³ De acordo com as políticas europeias de privacidade, dados coletados por câmeras de vídeo não podem ser divulgados sem anonimização prévia, removendo qualquer informação que permita identificar indivíduos diretamente.

⁴ O SHA-512 (*Secure Hash Algorithm* de 512 bits) pertence à família SHA-2, desenvolvida pela NSA e padronizada pelo NIST. A função produz um valor de hash fixo de 512 bits a partir de uma entrada arbitrária. Especificação disponível em: <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>>.

⁵ <<https://www.gifrangroup.it>>

⁶ <<https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier>>

resultados.

A abordagem proposta apresenta contribuições em quatro eixos principais: (i) análise de densidade do tráfego e identificação de congestionamentos; (ii) reconhecimento automático de placas de veículos; (iii) classificação do tipo de veículo, permitindo análises comportamentais diferenciadas; e (iv) desenvolvimento de algoritmos integrados à infraestrutura de mobilidade urbana, com vistas a um modelo sustentável de gestão inteligente. Segundo os autores, a combinação entre ALPR e *YOLO* aumenta a precisão e a robustez do sistema, oferecendo confiabilidade para uso contínuo em ambientes urbanos dinâmicos.

Os resultados indicam que o sistema alcança alta precisão na detecção e discriminação de veículos em tempo real, operando de forma eficiente em plataformas de borda com posterior transmissão dos dados anonimizados para servidores em nuvem via conexão LTE 4G. A integração entre detecção rápida e anonimização garante conformidade legal sem comprometer o desempenho, tornando a solução adequada para aplicações urbanas de larga escala. Embora o artigo mencione comparações com outros algoritmos, os dados apresentados são provenientes de estudos externos, o que restringe a avaliação direta de desempenho.

Entre as limitações, destacam-se a falta de análise aprofundada sobre condições adversas, como chuva intensa, contraluz, imagens noturnas ou oclusões, além da ausência de testes com câmeras móveis ou cenários de dupla mobilidade. O sistema também não realiza rastreamento individualizado de veículos ao longo do tempo, focando exclusivamente em análises agregadas de fluxo. Outro ponto é a impossibilidade de associar veículos específicos a trajetórias urbanas, uma vez que os dados são anonimizados logo após o reconhecimento da placa.

Para trabalhos futuros, os autores sugerem aprimorar a precisão em condições ambientais variadas, ampliar o número de tipos de veículos reconhecidos e desenvolver algoritmos de previsão de tráfego com base nos dados coletados. Tais direções de pesquisa estão alinhadas com tendências globais na área de mobilidade inteligente, reforçando o papel da visão computacional em sistemas urbanos distribuídos.

Em comparação com a proposta deste trabalho, observa-se que, embora Spanu et al. integrem detecção, identificação e classificação de veículos em uma única arquitetura, seu sistema não tem como objetivo o rastreamento veicular individualizado, tampouco incorpora sensores móveis ou câmeras transportadas em veículos de usuários. O *SmarTTrack*, por sua vez, avança no sentido de integrar OCR, rastreamento multiveicular em tempo real e captura por câmeras móveis, ao mesmo tempo em que se conecta a uma plataforma urbana padronizada (*InterSCity*). Assim, enquanto o trabalho de Spanu et al. contribui para análises de mobilidade urbana em larga escala, o *SmarTTrack* foca na identificação individualizada de veículos, com aplicações diretas para segurança pública e monitoramento

colaborativo.

3.4 Vehicle Route Tracking System Based on Vehicle Registration Number Recognition Using Template Matching Algorithm

Kiew et al. (KIEW et al., 2021) apresentam um sistema de rastreamento de rotas veiculares baseado no reconhecimento de placas utilizando a técnica de *template matching*. O trabalho parte da premissa de que o reconhecimento automático das placas pode subsidiar mecanismos simples de rastreamento, permitindo identificar a rota percorrida por um veículo a partir da sequência de câmeras pelas quais ele transita. Embora se trate de uma abordagem de baixo custo computacional, voltada a ambientes com infraestrutura limitada, os autores elaboram uma solução completa que integra OCR, banco de dados e visualização de trajetórias.

O sistema proposto funciona a partir da análise de imagens provenientes de câmeras de vigilância conectadas em rede. A arquitetura geral inclui dois componentes principais: um módulo de OCR, responsável por detectar e reconhecer os caracteres da placa, e um módulo de rastreamento, que organiza as detecções ao longo do tempo e as converte em uma rota exibida em mapa. A etapa de OCR emprega o *template matching*, que consiste em comparar cada caractere segmentado com um conjunto de caracteres-modelo previamente armazenados em banco de dados. O pipeline inclui segmentação da placa, binarização, identificação de regiões contendo caracteres, detecção de bordas, normalização geométrica (*deskew*), segmentação fina dos caracteres e, por fim, o processo de reconhecimento baseado na melhor correspondência obtida com os templates disponíveis.

Os autores também comparam o desempenho do *template matching* com redes neurais artificiais, destacando que, para o conjunto de imagens analisado, o *template matching* apresentou maior acurácia e menor tempo de processamento. Em função disso, a escolha do método se justifica pela necessidade de um processo rápido e leve, adequado ao hardware utilizado. O sistema integra ainda a ferramenta OPENALPR, que também utiliza *template matching* em sua etapa de reconhecimento.

A metodologia é dividida em duas fases: treinamento e implementação. Na fase de treinamento, o modelo foi ajustado especificamente para placas da Malásia, utilizando um conjunto de 310 imagens. O treinamento abrange tanto o detector de placas — baseado na marcação manual das regiões de interesse, conforme exibido nas figuras do artigo — quanto o OCR, treinado com caracteres segmentados e organizados em folhas de treinamento do Tesseract. Na fase de implementação, o sistema é executado em um ambiente Linux, onde imagens são processadas e registradas em um banco de dados. A partir das detecções, um módulo desenvolvido em *ASP.NET* integra as informações com a API do Google Maps, gerando a rota percorrida pelo veículo com base nas câmeras pelas quais ele passou.

Os experimentos foram realizados em um cenário real utilizando dez câmeras fixas, cujas localizações geográficas são fornecidas pelos autores. Os resultados mostram que o sistema foi capaz de reconhecer as placas com uma acurácia média em torno de 80% e reconstruir corretamente as rotas de diversos veículos testados, exibindo a trajetória e a posição atual estimada no mapa. Os autores ressaltam que fatores como clima, ruído nas imagens e variações de nitidez afetam o desempenho do OCR, enquanto o mecanismo de rastreamento apresentou resultados consistentes.

Apesar de suas contribuições, o sistema apresenta algumas limitações. Ele depende integralmente de câmeras fixas e não opera sobre fluxos contínuos de vídeo em tempo real, realizando apenas detecção quadro a quadro. O rastreamento é reconstruído a partir da ordem das câmeras que detectam o mesmo veículo, e não por um mecanismo de identificação contínua entre quadros. Ademais, técnicas como *template matching* tendem a ser sensíveis a ruídos, distorções e ângulos não padronizados, exigindo condições controladas para melhor desempenho. Ainda assim, trata-se de uma solução funcional para cenários com infraestrutura limitada e com requisitos modestos de processamento.

Em comparação com a abordagem desenvolvida neste trabalho, nota-se que o sistema de Kiew et al. possui escopo mais restrito, voltado para rastreamento básico em ambientes com câmeras fixas e sob condições relativamente controladas. Diferentemente, o *SmarTTrack* opera em cenários dinâmicos, com câmeras móveis, sob dupla mobilidade e utilizando modelos de detecção baseados em aprendizado profundo. Entretanto, ambos os sistemas compartilham a premissa de que informações extraídas das placas podem ser utilizadas para reconstruir trajetórias e apoiar aplicações de segurança ou gestão urbana — ainda que por mecanismos metodológicos distintos. Dessa forma, o trabalho de Kiew et al. oferece um ponto de partida conceitual relevante, especialmente ao demonstrar como pipelines simples de OCR podem ser integrados a sistemas maiores de monitoramento urbano.

3.5 Real-Time Multiple Vehicle Detection Using a Rear Camera Mounted on a Vehicle

Oheka e Tu (OHEKA; TU, 2018) apresentam um sistema de detecção e rastreamento de múltiplos veículos em tempo real utilizando uma câmera instalada na parte traseira de um veículo (Figura 5). A proposta parte da premissa de que, diante do aumento da complexidade do tráfego urbano e da crescente demanda por sistemas de assistência ao condutor, soluções de visão computacional podem contribuir para a prevenção de incidentes, como colisões, congestionamentos e dirigibilidade insegura. O sistema captura continuamente a cena posterior ao veículo hospedeiro e processa os quadros para identificar veículos que se aproximam, ativando alertas diante de situações potencialmente perigosas.

A solução proposta emprega técnicas clássicas de visão computacional combinadas com aprendizado de máquina supervisionado. A arquitetura do sistema é organizada em cinco etapas principais: (1) preparação dos dados, (2) pré-processamento, (3) segmentação do primeiro plano, (4) detecção de veículos e (5) rastreamento. Inicialmente, o vídeo é convertido em quadros individuais, sobre os quais são aplicados procedimentos de normalização de histograma para mitigar variações de iluminação, bem como filtragem de ruído por meio de filtro da mediana. Em seguida, realiza-se a segmentação por subtração de fundo, técnica amplamente empregada para separar objetos em movimento do cenário estático. Embora métodos convencionais de subtração de fundo sejam inadequados para câmeras móveis, os autores contornam essa limitação restringindo a análise a uma região de interesse (ROI) pré-determinada do quadro, na qual é mais provável que veículos apareçam, reduzindo a variabilidade do cenário e o custo computacional.

A detecção dos veículos é realizada por meio de um classificador em cascata baseado em características de Haar e no algoritmo AdaBoost, conforme descrito nos trabalhos clássicos de Viola e Jones ([VIOLA; JONES, 2004](#)). As características de Haar fornecem uma representação eficiente e de baixo custo computacional, enquanto o AdaBoost ([FREUND; SCHAPIRE, 1997](#)) é responsável por combinar diversos classificadores fracos em um modelo mais robusto. Os autores utilizam três estágios no classificador em cascata e complementam o treinamento realimentando o conjunto de amostras positivas com imagens capturadas pelo próprio sistema, buscando melhorar a precisão da detecção em cenários reais. As características de Haar fornecem descritores visuais simples baseados em diferenças de intensidade entre regiões retangulares adjacentes, sendo amplamente empregadas em sistemas de vigilância e monitoramento veicular ([TANG; XU; ZHANG, 2014](#)).

O *dataset* aplicado foi composto por imagens capturadas pela própria câmera traseira do veículo durante deslocamentos em áreas urbanas e rurais. O conjunto inclui 560 amostras positivas contendo veículos e 1000 amostras negativas, ampliadas para 1120 imagens positivas após espelhamento por simetria. Dois terços desse conjunto foram utilizados para treinamento e o restante para validação. Cada veículo detectado é rotulado como “carro” e delimitado por uma caixa retangular. O artigo, entretanto, não disponibiliza o conjunto de dados nem detalha a distribuição das condições ambientais presentes nos vídeos, como iluminação, clima ou densidade de tráfego.

Os resultados obtidos demonstram que o sistema alcança boas taxas de verdadeiros positivos e verdadeiros negativos, operando a aproximadamente 20 FPS, com tempo médio de 51 ms por quadro. A proposta mostrou-se eficaz em diferentes tipos de veículos — incluindo carros, caminhões e ônibus — ainda que alguns desafios tenham sido observados, como variações abruptas de iluminação, diferenças na silhueta entre modelos de veículos e dificuldades ocasionadas por oclusões. Os autores reconhecem essas limitações e sugerem, como trabalhos futuros, melhorias nos algoritmos de detecção para lidar com condições



Figura 5 – Veículo equipado com câmera traseira. Fonte: (OHEKA; TU, 2018)

ambientais adversas, bem como a investigação de técnicas mais modernas de segmentação e aprendizado de máquina para aumentar a robustez do sistema.

Embora o sistema apresente uma solução funcional, leve e de baixo custo para detecção traseira de veículos, seu escopo é restrito: não realiza reconhecimento de placas, não emprega técnicas de rastreamento multiveicular avançadas, não opera com câmeras móveis em dupla mobilidade e não integra os resultados a plataformas de Cidades Inteligentes. Em contraste, o *SmarTTrack* combina detecção, rastreamento, reconhecimento de placas e integração com a plataforma *InterSCity*, ampliando sua aplicabilidade em cenários urbanos dinâmicos que demandam identificação veicular completa e persistência estruturada dos eventos.

Embora o sistema apresente uma solução funcional, leve e de baixo custo para detecção traseira de veículos, sua concepção impõe limitações relevantes ao sensoriamento móvel participativo. A utilização de uma câmera posicionada exclusivamente na parte traseira do veículo restringe o campo de observação e reduz o potencial colaborativo do modelo, dificultando sua aplicação em cenários urbanos amplos e dinâmicos.

Nesse sentido, a abordagem proposta nesta dissertação adota uma concepção distinta, com câmeras posicionadas na parte frontal dos veículos (modo cidadão) ou fixadas à margem de rodovias (modo *blitz*), orientada ao monitoramento participativo em ambiente urbano real. O modelo realiza a detecção e o rastreamento veicular a partir da identificação das placas, com posterior integração a uma plataforma de Cidades Inteligentes.

3.6 Programmed Number Plate Recognition using Deep Learning

Ananthi et al. (ANANTHI et al., 2023) apresentam o sistema PNPR, uma abordagem de reconhecimento de placas veiculares baseada em aprendizado profundo que integra diferentes técnicas de visão computacional, segmentação morfológica e redes neurais convolucionais. O *pipeline* (Figura 6) inicia-se com a aquisição das imagens,

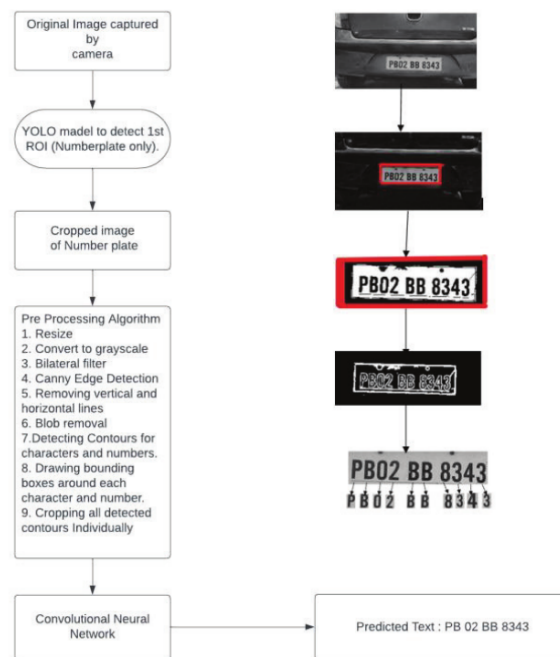


Figura 6 – Arquitetura do PNPR. Fonte: (ANANTHI et al., 2023)

seguida da detecção da região da placa, que é realizada por meio do algoritmo *YOLO*, responsável por identificar a área de interesse e recortar a placa para as fases posteriores do processamento. Em seguida, se dá o pré-processamento, que inclui técnicas, como limiarização, aprimoramento de imagem e redução de ruído, conforme detalhado pelos autores.

Após a localização da placa veicular, o método realiza a segmentação dos caracteres a partir da obtenção de uma imagem binarizada da região de interesse. Em seguida, são aplicados procedimentos de recorte e normalização, permitindo isolar individualmente cada caractere presente na placa. O reconhecimento é então realizado por meio de uma arquitetura híbrida, que combina redes neurais convolucionais (CNNs) para a extração de padrões visuais com uma rede LSTM. Esta rede é empregada para modelar a dependência sequencial entre os caracteres, considerando a ordem e o contexto em que aparecem na placa, o que contribui para a formação consistente da sequência final de caracteres reconhecidos.

O ambiente de implementação do sistema utiliza *Python*, *OpenCV*, *NumPy* e *TensorFlow*, executado em hardware de baixo custo (CPU i5, 8 GB RAM, GPU de 4 GB). O *dataset* empregado reúne 10.000 imagens de placas capturadas em diferentes condições ambientais, incluindo variações de iluminação e clima. Segundo os resultados apresentados pelos autores, o sistema atinge 97,6% de acurácia na detecção da placa, 97,8% na segmentação e 96% no reconhecimento dos caracteres, valores consolidados nas tabelas experimentais presentes no trabalho.

Apesar da elevada precisão, o PNPR apresenta limitações estruturais relevantes. O método opera majoritariamente sobre imagens estáticas, não implementa rastreamento multiveicular, não processa vídeo contínuo em tempo real e não considera cenários com câmeras móveis. Além disso, não realiza integração com plataformas de Cidades Inteligentes, restringindo sua aplicabilidade a contextos controlados de ALPR. Os autores também reconhecem a necessidade de ampliar os datasets e avaliar o modelo sob condições mais adversas, incluindo maior diversidade de ângulos e padrões de placas.

Quando comparado ao *SmarTTrack*, observa-se que o PNPR concentra-se exclusivamente no reconhecimento de placas, enquanto o sistema proposto neste trabalho integra detecção, rastreamento, OCR e operação sob dupla mobilidade em fluxos contínuos de vídeo. Ademais, o *SmarTTrack* conecta-se à plataforma *InterSCity*, permitindo a persistência e a consulta unificada dos eventos detectados, o que amplia significativamente a utilidade do sistema em aplicações reais de segurança pública. Nesse sentido, embora o PNPR represente uma solução precisa e robusta para ALPR em imagens isoladas, sua aplicabilidade permanece limitada frente às demandas de monitoramento urbano dinâmico e participativo.

3.7 License Plate Recognition in Urban Road Based on Vehicle Tracking and Result Integration

Zhu et al. (ZHU et al., 2019) apresentam um sistema de Reconhecimento Automático de Placas (ALPR) projetado especificamente para lidar com vídeos de baixa qualidade provenientes de ambientes urbanos. A proposta busca superar limitações tradicionais de segmentação, recorrendo a uma estratégia de reconhecimento *end-to-end* por meio de redes neurais convolucionais, complementada por um mecanismo de integração de resultados ao longo dos quadros do vídeo (Figura 7). O objetivo central é melhorar a robustez do reconhecimento sob condições adversas, como trepidações, baixa resolução e variações bruscas de iluminação, desafios comuns em sistemas de vigilância de tráfego. O fluxo de processamento inicia-se com a detecção de veículos, etapa seguida pelo rastreamento contínuo ao longo do vídeo. Uma vez identificado e estabilizado o alvo, o sistema aplica um detector baseado em SSD para localizar a região da placa, evitando técnicas tradicionais de segmentação manual ou heurística. Em vez de realizar o reconhecimento caractere a caractere, a abordagem propõe uma CNN treinada para reconhecer a placa completa em uma única etapa, caracterizando uma solução *end-to-end*. Para mitigar as oscilações entre quadros, os autores implementam um mecanismo de integração temporal dos resultados, que combina as predições utilizando votação ponderada e probabilidades derivadas de *softmax*. Essa estratégia permite reduzir erros ocasionais em quadros específicos, aumentando a confiabilidade do reconhecimento final.



Figura 7 – Estrutura do sistema LPR baseado no rastreamento de veículos e na integração de resultados. Fonte: (ZHU et al., 2019)

O trabalho foi avaliado em três conjuntos de dados distintos, incluindo ambientes com condições de Alto Alcance Dinâmico (HDR) e cenários voltados para reidentificação de veículos (*ReId*). Os resultados relatados indicam acurácia de 62,21% em ambiente HDR e 89,80% em contexto de *ReId*, demonstrando que a combinação de rastreamento com integração temporal melhora significativamente o desempenho em comparação com a análise isolada de quadros. Embora robusto para vídeos estáticos, o sistema foi projetado para câmeras fixas, não abordando situações com dupla mobilidade, nas quais tanto o veículo quanto a câmera estão em movimento simultâneo.

As limitações da abordagem incluem dependência de vídeos com campo de visão estável, ausência de um módulo de OCR especializado para diferentes padrões regionais de placas e falta de integração com plataformas urbanas ou sistemas baseados em Cidades Inteligentes. Os autores também não apresentam mecanismos de anonimização, persistência de dados ou interoperabilidade com serviços externos.

Em relação ao *SmarTTrack*, o trabalho de Zhu et al. é relevante por demonstrar como técnicas de rastreamento podem estabilizar o reconhecimento de placas ao longo do tempo. Entretanto, enquanto o método analisado opera exclusivamente sobre câmeras estáticas e não realiza identificação multiveicular integral, o *SmarTTrack* incorpora detecção, rastreamento, OCR e integração direta com a plataforma *InterSCity*, além de operar sob condições de dupla mobilidade típicas do *mobile crowdsensing*. Dessa forma, embora o estudo contribua para compreender estratégias de integração temporal de predições, sua aplicabilidade permanece limitada frente às demandas de monitoramento colaborativo e

dinâmico adotadas nesta dissertação.

3.8 Discussão Comparativa e Síntese

A Tabela 1 sintetiza os principais aspectos comparados entre os trabalhos. Observa-se que nenhum deles integra, simultaneamente: rastreamento multiveicular em tempo real, identificação automática de placas, uso de câmeras móveis, operação sob dupla mobilidade e integração com uma plataforma urbana padronizada. O *SmarTTrack* propõe justamente essa combinação, preenchendo lacunas relevantes e contribuindo para aplicações práticas em segurança pública e monitoramento urbano.

Tabela 1 – Comparação entre os sistemas analisados

Características	(CHEN et al., 2019)	(SPANU et al., 2021)	(KIEW et al., 2021)	(OHEKA; TU, 2018)	(ANANTHI et al., 2023)	(ZHU et al., 2019)	SmarTTrack
Usa Crowdsensing	✓						✓
Processa Vídeo em Tempo Real		✓		✓			✓
Detecta Veículos	✓	✓		✓	✓	✓	✓
Rastreia Veículos	✓	✓	✓	✓		✓	✓
Extraí Placas		✓	✓		✓	✓	✓
Integra Smart City		✓					✓
Usa Câmera Móvel	✓			✓			✓

3.9 Considerações Finais

A análise dos trabalhos relacionados evidenciou avanços importantes em rastreamento veicular, reconhecimento de placas e monitoramento urbano por visão computacional. No entanto, as abordagens identificadas tratam esses elementos de forma parcial, sem integrar, de maneira simultânea, sensoriamento móvel participativo, rastreamento multiveicular em tempo real, reconhecimento automático de placas, operação sob dupla mobilidade e integração com uma plataforma de Cidades Inteligentes.

Nesse contexto, o *SmarTTrack* distingue-se por reunir esses componentes em uma única arquitetura, direcionada ao monitoramento veicular em cenários urbanos reais. Assim, este capítulo permitiu situar a proposta no estado da arte, evidenciar suas lacunas e justificar a relevância da solução desenvolvida, cuja metodologia e arquitetura são detalhadas no capítulo seguinte.

4 Metodologia

Este capítulo descreve de forma detalhada a metodologia adotada para o desenvolvimento e a avaliação de um modelo de rastreamento veicular voltado a Cidades Inteligentes, fundamentado nos princípios de Sensoriamento Móvel Participativo. A proposta metodológica foi concebida com o objetivo de permitir a identificação, o rastreamento e o reconhecimento de placas veiculares em ambientes urbanos reais, explorando a participação colaborativa de cidadãos por meio do uso de dispositivos móveis amplamente disponíveis, como *smartphones*.

No modelo proposto, os usuários utilizam um aplicativo móvel para capturar vídeos do ambiente viário, os quais são posteriormente processados por técnicas de visão computacional e aprendizado profundo com o intuito de identificar veículos de interesse das autoridades de segurança pública. Essa abordagem busca ampliar a cobertura espacial do monitoramento urbano sem a necessidade de investimentos adicionais em infraestrutura fixa, alinhando-se aos princípios de escalabilidade, baixo custo e participação cidadã característicos de soluções para Cidades Inteligentes.

A metodologia foi estruturada de modo a garantir rigor científico, reprodutibilidade experimental e clareza na definição das decisões de projeto. Para isso, o processo metodológico foi organizado em etapas bem definidas, que abrangem desde a aquisição

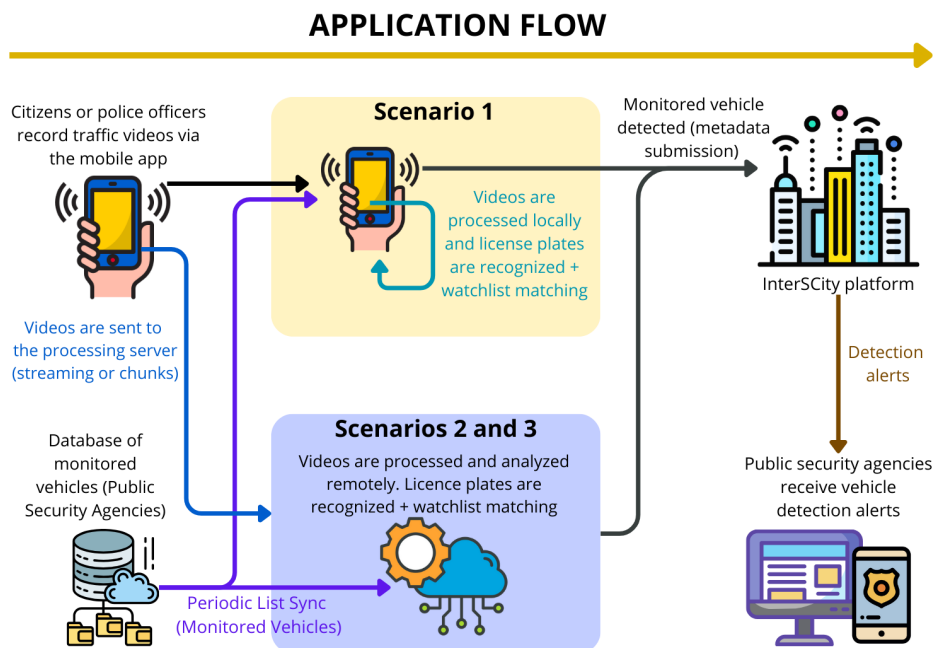


Figura 8 – Arquitetura Geral do *SmartTrack*. Fonte: o autor

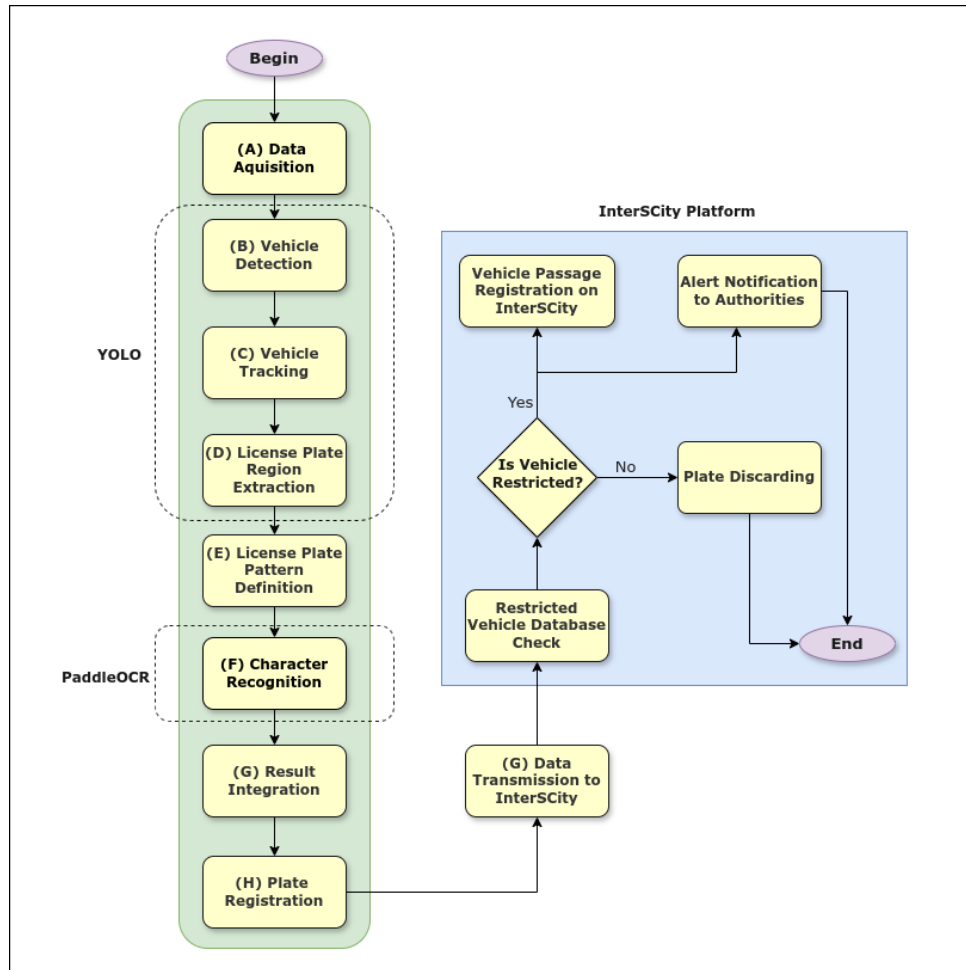


Figura 9 – SmarTTack Pipeline. Fonte: o autor

dos dados em cenários reais de tráfego até a integração dos resultados em uma plataforma urbana de gerenciamento. A arquitetura geral do sistema proposto, denominado *SmarTTack*, é apresentada na Figura 8, enquanto o fluxo operacional completo do processamento — incluindo detecção, rastreamento, reconhecimento de placas e consolidação dos resultados — é ilustrado na Figura 9.

Adicionalmente, a metodologia considera explicitamente diferentes estratégias de processamento, contemplando tanto a execução local em dispositivos móveis quanto o processamento remoto em servidores externos. Essa distinção permite analisar o impacto de restrições computacionais, consumo de recursos e latência na execução do sistema, fornecendo subsídios para uma avaliação comparativa entre cenários e preparando o terreno para os experimentos descritos nos capítulos subsequentes.

Nas seções seguintes, cada uma das etapas do método proposto é apresentada em detalhe, com a explicitação das decisões metodológicas adotadas, dos parâmetros empregados e das tecnologias utilizadas em cada fase do sistema.

4.1 Aquisição dos Dados

A aquisição dos dados é realizada por meio de câmeras de *smartphones*, explorando a ampla disponibilidade desses dispositivos como sensores móveis em ambientes urbanos. Os aparelhos podem ser utilizados de duas formas distintas: (i) embarcados em veículos, fixados em suportes frontais, caracterizando o modo cidadão (Figura 10) ou (ii) posicionados em tripés em pontos estratégicos à margem de vias e rodovias, caracterizando o modo *blitz*. Este último é destinado, por exemplo, à atuação de agentes de segurança pública em operações policiais ou ações de fiscalização viária.

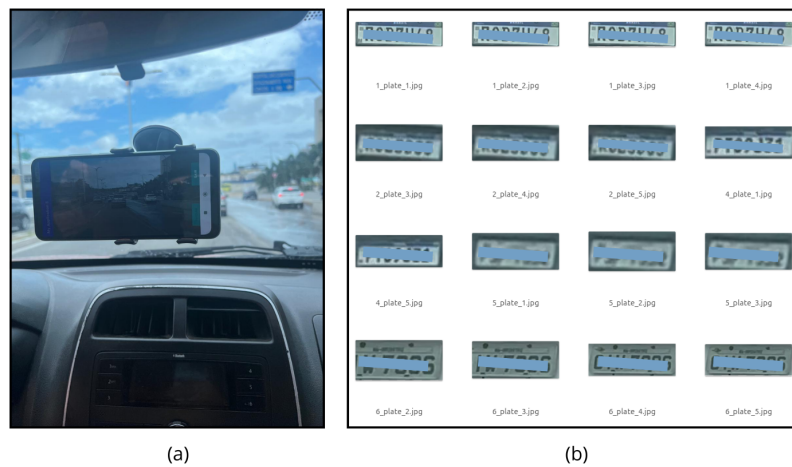


Figura 10 – (a) Aplicativo *SmarTTrack* utilizado em ambiente real no modo cidadão, em suporte veicular fixo. (b) Exemplos de placas veiculares extraídas automaticamente pelo sistema. Por questões de privacidade e proteção de dados, as informações alfanuméricas das placas foram propositalmente borradas nas imagens apresentadas. Fonte: o autor

Em ambos os modos de operação, as câmeras capturam vídeos em condições reais de tráfego, sujeitos a variações significativas de iluminação, ângulo de incidência, oclusões parciais e diferentes resoluções. Essas características conferem ao conjunto de dados uma elevada diversidade visual, refletindo cenários urbanos não controlados e mais próximos das situações encontradas em aplicações reais de monitoramento veicular.

A adoção dessa estratégia de aquisição amplia a cobertura geográfica do sistema proposto, permitindo o monitoramento de regiões não atendidas por infraestruturas fixas de vigilância, como câmeras estáticas ou sistemas embarcados dedicados. Além disso, ao possibilitar a participação ativa dos cidadãos no processo de coleta de dados, o modelo proposto fortalece o paradigma de Sensoriamento Móvel Participativo, contribuindo para ações de segurança pública em Cidades Inteligentes sem a imposição de custos adicionais ao Estado.

4.2 Treinamento e Validação do Detector de Placas

O detector de placas veiculares empregado neste trabalho foi desenvolvido a partir da arquitetura *YOLOv11*, utilizando a variante intermediária *YOLOv11m*. A escolha dessa configuração fundamenta-se no equilíbrio entre capacidade de generalização, acurácia de detecção e custo computacional, aspecto relevante considerando a possibilidade de futuras adaptações do modelo para execução em dispositivos móveis ou em ambientes com restrições de recursos. O processo de treinamento teve como objetivo principal capacitar o modelo a identificar, de forma robusta, regiões correspondentes a placas veiculares em cenários urbanos reais. Para atender a esses requisitos, foi adotada a seguinte configuração experimental:

- **Modelo base:** *yolov11m.pt*;
- **Número de épocas:** 130;
- **Resolução de entrada:** 640×640 pixels;
- **Tamanho do lote:** 16;
- **Número de classes:** 1 (placa veicular).

O conjunto de dados utilizado no treinamento foi o **UFPR-ALPR**¹ (Laroca et al., 2018), reconhecido na literatura como um conjunto de dados adequado ao treinamento e à avaliação de modelos de reconhecimento automático de placas veiculares em ambientes não controlados. O *dataset* é composto por aproximadamente 4.500 imagens totalmente anotadas, capturadas em cenários reais de tráfego, nos quais tanto o veículo monitorado quanto o veículo que hospeda a câmera encontram-se em movimento. As imagens foram adquiridas majoritariamente com câmeras *GoPro Hero4 Silver* e abrangem cerca de 150 veículos distintos, contemplando diferentes categorias, fabricantes, modelos e anos de fabricação.

Cada amostra do *dataset* inclui a imagem do veículo, as coordenadas dos quatro vértices da placa, a localização individual dos caracteres e metadados adicionais, como marca, modelo e ano do veículo. Para viabilizar o uso desse conjunto de dados no treinamento do *YOLOv11*, foi necessária a conversão das anotações originais para o formato compatível com o modelo, a partir da extração da caixa delimitadora mínima que engloba a região da placa.

Após a conversão, as imagens e respectivas anotações foram reorganizadas segundo a estrutura padrão exigida pelo *YOLOv11*, com separação explícita entre os conjuntos de treinamento (*train*), validação (*val*) e teste (*test*). Foi mantida a proporção original definida

¹ <<https://github.com/raysonlaroca/ufpr-alpr-dataset>>

pelos autores do conjunto de dados *UFPR-ALPR*, correspondente a 40% das amostras destinadas ao treinamento, 40% ao teste e 20% à validação. Essa organização possibilitou a avaliação consistente do desempenho do modelo durante o processo de treinamento e a análise de sua capacidade de generalização para outros conjuntos de dados.

A avaliação do detector foi realizada separadamente nos conjuntos de treinamento e validação, utilizando como métricas de desempenho a precisão, a revocação e o *F1-score*, amplamente adotadas na literatura para avaliar a eficácia de sistemas de detecção de objetos. No conjunto de treinamento, o modelo apresentou precisão de 99,3%, revocação de 97,7% e *F1-score* de 98,5%. Já no conjunto de validação, utilizado para avaliar a capacidade de generalização do modelo, foram obtidos precisão de 100,0%, revocação de 96,9% e *F1-score* de 98,4%.

A Figura 11 apresenta a matriz de confusão normalizada e as métricas consolidadas do processo de treinamento e validação do detector de placas baseado no *YOLOv11m*.

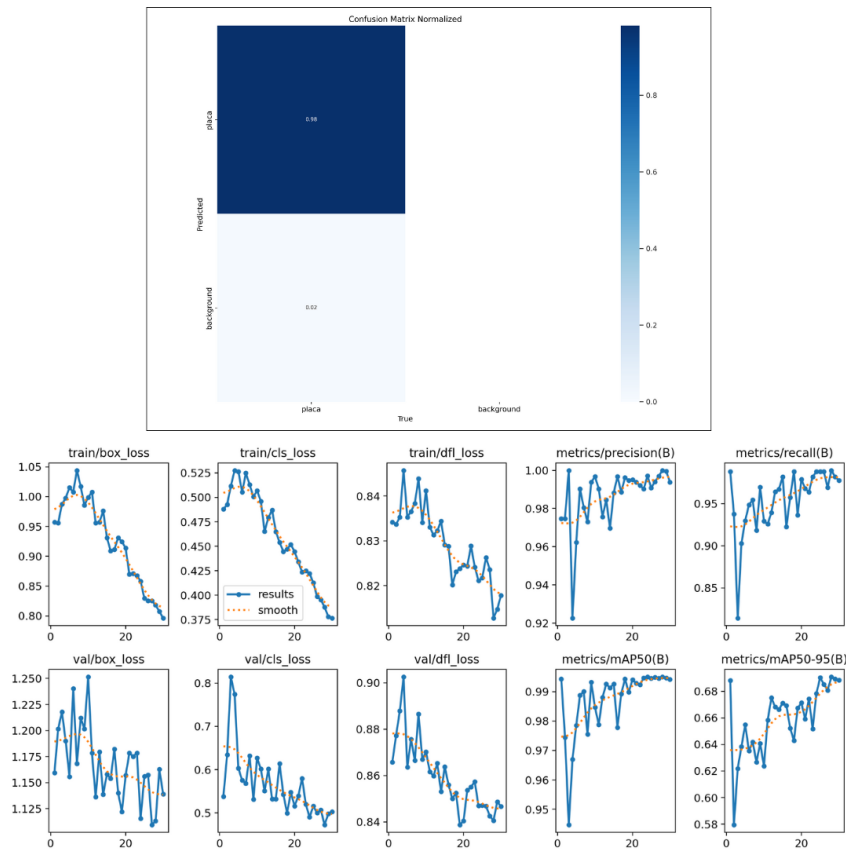


Figura 11 – Matriz de confusão e métricas de desempenho do detector de placas baseado no *YOLOv11m*. Fonte: o autor

Os resultados obtidos indicam um desempenho elevado do modelo tanto no conjunto de treinamento quanto no de validação, evidenciando um equilíbrio consistente entre precisão e revocação. Os elevados valores de *F1-score* confirmam a robustez do detector, assegurando que a maioria das placas presentes nas imagens seja corretamente

identificada, com baixa incidência de falsos positivos e falsos negativos. Esse desempenho é fundamental para as etapas subsequentes do sistema proposto, uma vez que a qualidade da detecção influencia diretamente a eficiência do rastreamento e do reconhecimento óptico de caracteres.

4.3 Detecção de Veículos por meio de suas placas

A detecção de veículos no sistema proposto é realizada por meio da rede neural convolucional *YOLO* previamente treinada para este fim. Como já explicitado, o *YOLO* é uma das arquiteturas mais consolidadas na literatura para detecção de objetos em tempo real e caracteriza-se por tratar a tarefa de detecção como um problema de regressão direta, permitindo a identificação simultânea de múltiplos objetos em uma única passagem pela rede, o que resulta em baixa latência e alta eficiência computacional (REDMON et al., 2016).

No contexto deste trabalho, a arquitetura *YOLOv11* é empregada em diferentes etapas do *pipeline*, desempenhando papel central na detecção e, quando viável, no rastreamento de veículos a partir da identificação e do acompanhamento das regiões de placa. Embora o objetivo final seja o rastreamento de veículos, a associação temporal entre os objetos ao longo do vídeo é realizada com base na localização da placa, que atua como elemento discriminante e relativamente estável durante o processo. O modelo é ainda utilizado na segmentação e recorte das regiões de interesse (ROI) correspondentes às placas, contribuindo para a padronização do processamento e para a redução da complexidade do sistema.

Além disso, a eficiência computacional do *YOLOv11* permite sua adoção em diferentes cenários operacionais do *SmartTrack*. No dispositivo móvel, o algoritmo é utilizado exclusivamente como detector, pois o rastreamento em tempo real com o módulo integrado ao *YOLO* impõe custo computacional elevado para *smartphones*. Nessa configuração, o rastreamento é delegado a métodos probabilísticos leves, baseados no Filtro de Kalman, na etapa de processamento local. Em contrapartida, no servidor remoto é possível ativar o módulo de rastreamento do *YOLO*, permitindo acompanhar a trajetória das placas ao longo da sequência sem comprometer o desempenho.

4.4 Rastreamento de Veículos e Extração da Área da Placa

Nesta seção, descreve-se o núcleo do processamento de visão computacional do *SmartTrack*: o rastreamento de veículos ao longo do fluxo de vídeo e a extração das regiões de interesse correspondentes às placas veiculares. Essas etapas são fundamentais para sustentar, nas seções seguintes, o reconhecimento óptico de caracteres e a consolidação

da identificação final de cada veículo, especialmente em contextos urbanos dinâmicos e típicos de Cidades Inteligentes.

Considerando a heterogeneidade de dispositivos móveis e as limitações de conectividade inerentes a uma abordagem baseada em Sensoriamento Móvel Participativo (SMP), a solução foi avaliada em três cenários experimentais. No primeiro, todo o processamento é realizado localmente no dispositivo móvel. Nos demais, o dispositivo atua como coletor, e o processamento é executado remotamente, diferindo apenas na estratégia de envio do vídeo: transmissão contínua e transmissão segmentada em blocos (*chunks*). Assim, as subseções a seguir detalham como o rastreamento e a extração das placas são operacionalizados nas abordagens remota e local, evidenciando as principais decisões técnicas que visam manter identificadores consistentes ao longo do tempo e obter recortes adequados para o reconhecimento posterior.

4.4.1 Rastreamento remoto

Nos cenários de processamento remoto, o vídeo capturado pelo dispositivo móvel é transmitido para um servidor externo, onde o processamento é realizado de forma centralizada. Nessa abordagem, o dispositivo atua como coletor, enquanto a análise é executada remotamente, podendo ocorrer por transmissão contínua do vídeo ou por envio segmentado em trechos. Nesse contexto, o rastreamento de veículos é executado de maneira integrada à etapa de detecção, utilizando os mecanismos nativos de rastreamento disponibilizados pelo *YOLOv11*. Cada veículo detectado recebe um identificador único, o qual é mantido ao longo dos quadros sucessivos, possibilitando o acompanhamento contínuo de sua trajetória no fluxo de vídeo.

Após a detecção e o rastreamento dos veículos, as regiões correspondentes às placas veiculares são extraídas a partir das caixas delimitadoras associadas aos objetos rastreados. Para essa finalidade, é utilizado o detector de placas treinado com a arquitetura *YOLOv11*, capaz de identificar placas de diferentes categorias de veículos, incluindo automóveis, ônibus, caminhões e motocicletas, cujos formatos e disposições de caracteres apresentam variações estruturais.

A Figura 10 ilustra exemplos de uso do aplicativo *SmarTTrack* em ambiente real, bem como amostras de placas veiculares extraídas automaticamente pelo sistema nos cenários de processamento remoto.

4.4.2 Rastreamento local

No cenário de processamento local, o rastreamento é realizado diretamente no dispositivo móvel. Devido às limitações computacionais e às restrições impostas pela plataforma *Android*, não é viável empregar, de forma consistente e no escopo desta

implementação, os mesmos mecanismos de rastreamento utilizados no processamento remoto. Assim, optou-se pela adoção de um método de rastreamento probabilístico leve baseado no Filtro de Kalman, amplamente utilizado para estimativa de estados em sistemas dinâmicos (KALMAN, 1960).

Nesse cenário, os quadros capturados pela câmera do dispositivo são processados de forma sequencial, sendo inicialmente submetidos à etapa de detecção de veículos por meio do *YOLOv11*. A partir das caixas delimitadoras obtidas, o Filtro de Kalman é empregado para estimar a posição futura dos objetos detectados e estabelecer a associação entre detecções consecutivas ao longo do tempo. Cada objeto rastreado recebe um identificador persistente, permitindo distinguir veículos recém-detectados daqueles que já estavam sendo acompanhados em quadros anteriores.

A associação entre as detecções atuais e as estimativas previstas pelo Filtro de Kalman é realizada com base na interseção entre as caixas delimitadoras, o que contribui para manter a continuidade do rastreamento diante de oclusões parciais, variações de posição ou perdas temporárias de detecção. Quando essa associação não é possível por múltiplos ciclos consecutivos, o identificador correspondente é considerado inativo, e o rastreamento do objeto é encerrado.

O fluxo de execução compreende as etapas de captura e amostragem de quadros, detecção de veículos, associação temporal das detecções por meio do Filtro de Kalman e extração das regiões de interesse (ROI) correspondentes às placas veiculares. Essas regiões são então encaminhadas às etapas subsequentes de reconhecimento óptico de caracteres e validação das sequências detectadas, conforme descrito nas seções seguintes.

4.5 Definição do Padrão da Placa

No Brasil, atualmente circulam dois tipos de placas veiculares: o padrão Mercosul e o antigo padrão brasileiro, este último já descontinuado e gradualmente substituído pelo primeiro. Para aprimorar a extração de caracteres e reduzir erros de reconhecimento óptico, torna-se essencial determinar qual padrão de placa está presente na imagem antes de aplicar os algoritmos de OCR. Para isso, utilizamos técnicas de Processamento Digital de Imagens implementadas com a biblioteca *OpenCV* e manipulação de arrays com *NumPy*, permitindo a segmentação e análise eficiente da região da placa.

O método implementado realiza a identificação do tipo de placa a partir da detecção da faixa azul característica do padrão Mercosul. Primeiramente, a imagem da placa é carregada e redimensionada para 400x150 *pixels*, a fim de otimizar o processamento e reduzir ruídos. Em seguida, a imagem é convertida para o espaço de cor HSV (*Hue*, *Saturation*, *Value*), que facilita a segmentação baseada em cores, uma vez que permite a distinção mais robusta de tons específicos em diferentes condições de iluminação.

Para isolar a faixa azul do padrão Mercosul, aplicamos uma máscara de cor utilizando a função `cv2.inRange()`, com limites definidos para tons de azul no espaço HSV. Essa faixa foi determinada empiricamente para maximizar a sensibilidade à cor azul enquanto minimiza falsos positivos. Após a segmentação, restringimos a análise a uma região central superior da imagem, que corresponde à área onde a faixa azul geralmente está presente nas placas do padrão Mercosul. A contagem de pixels não nulos (ou seja, pixels detectados como pertencentes à faixa azul) é então realizada com `cv2.countNonZero()`.

A decisão sobre o tipo de placa é feita com base na contagem desses pixels azuis. Se o número de pixels identificados ultrapassar um limite experimentalmente definido, a placa é classificada como padrão Mercosul (M); caso contrário, assume-se que a placa pertence ao padrão antigo (A). Esse limiar foi ajustado com base em testes empíricos para garantir alta acurácia na distinção entre os padrões. Ao identificar corretamente o tipo de placa, é possível aplicar a máscara de ajuste correta para a extração dos caracteres, potencializando a precisão do OCR ao restringir os caracteres válidos de acordo com a formatação esperada para cada padrão de placa.

4.6 Reconhecimento de Caracteres

Para a etapa de reconhecimento dos caracteres presentes nas placas veiculares, o sistema emprega a biblioteca *PaddleOCR*, uma solução de reconhecimento óptico de caracteres de código aberto desenvolvida no ecossistema PaddlePaddle. Essa ferramenta destaca-se pela sua robustez frente a variações de iluminação, resolução e qualidade visual, características comuns em cenários urbanos reais, além de oferecer suporte à execução eficiente em CPUs, GPUs e dispositivos móveis.

No presente trabalho, é utilizado o módulo *PP-OCR (Practical PaddleOCR)*, que adota uma arquitetura de dois estágios. No primeiro estágio, realiza-se a detecção das regiões textuais por meio do algoritmo *Differentiable Binarization (DB)*, responsável por segmentar áreas contendo texto de forma precisa, mesmo em condições adversas, como baixa iluminação, ruído visual ou desgaste físico das placas. No segundo estágio, o reconhecimento dos caracteres é conduzido por uma arquitetura baseada em *Convolutional Recurrent Neural Network (CRNN)*, capaz de identificar sequências de caracteres diretamente a partir das regiões detectadas, dispensando a segmentação explícita de cada símbolo individual.

Adicionalmente, o *PP-OCR* incorpora um mecanismo de classificação de orientação do texto, que ajusta automaticamente a direção das regiões analisadas antes da etapa de reconhecimento. Esse recurso contribui para a redução de erros decorrentes de inclinações, rotações ou pequenas distorções geométricas presentes nas imagens das placas veiculares, aumentando a confiabilidade do processo de leitura.

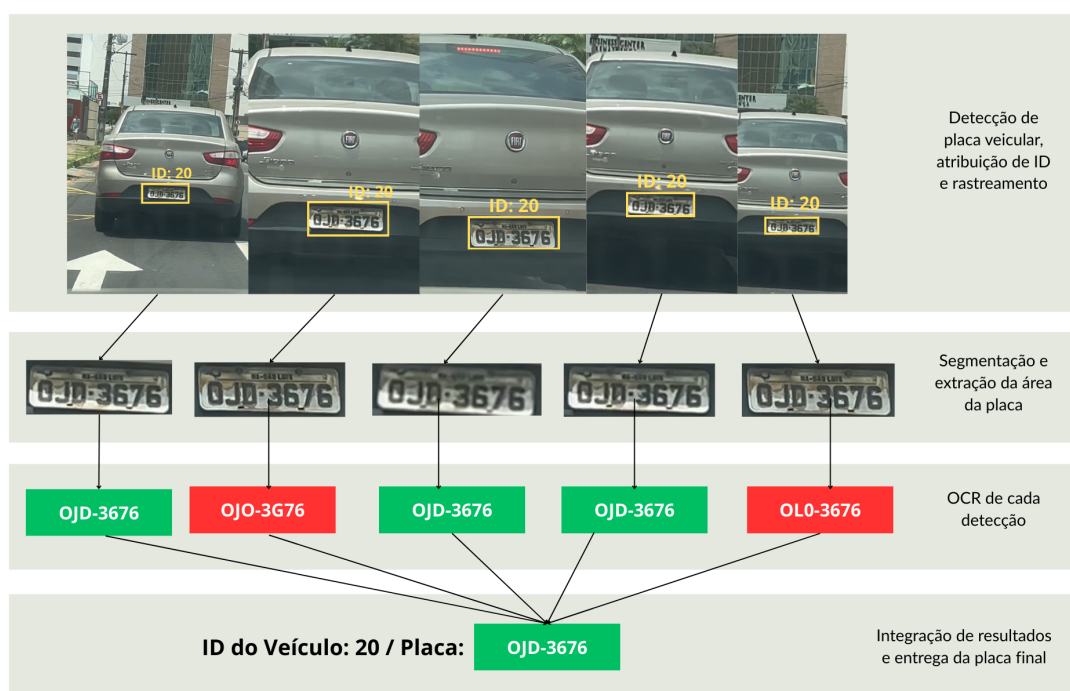


Figura 12 – Integração de resultados no *SmartTrack*. Fonte: o autor

4.7 Integração dos Resultados

Os resultados obtidos para cada placa extraída de um veículo identificado e rastreado por um *ID* são consolidados com o objetivo de aumentar a confiabilidade da identificação final, conforme ilustrado na Figura 12. Como o reconhecimento óptico de caracteres pode apresentar variações entre quadros consecutivos — em função de oclusões, desfoque, ângulo de captura ou condições de iluminação — o modelo não se baseia em uma única leitura isolada, mas na combinação de múltiplas observações realizadas ao longo do tempo.

Durante o processamento do vídeo, os quadros são analisados de forma amostrada, de modo a equilibrar custo computacional e robustez do reconhecimento. Em vez de processar todos os quadros capturados pela câmera, o modelo seleciona periodicamente quadros ao longo da permanência do veículo no campo de visão, garantindo múltiplas tentativas de leitura da placa associada a um mesmo *ID*.

Em seguida, para cada região de placa recortada pela rede *YOLO*, o *PaddleOCR* realiza a extração da sequência de caracteres. As leituras obtidas são então submetidas a um processo de validação sintática, no qual apenas sequências que atendem aos formatos oficiais de placas brasileiras são consideradas válidas, a saber: o padrão antigo (*AAA-0000*) e o padrão Mercosul (*AAA0A00*). Leituras incompletas, com número insuficiente de caracteres ou que não se enquadram em nenhum dos padrões definidos são automaticamente descartadas.

Após a coleta de todas as leituras válidas associadas a um mesmo veículo, é realizada uma consolidação temporal dos resultados. Nesse processo, a placa final atribuída ao veículo corresponde à sequência de caracteres mais frequentemente observada entre as leituras válidas disponíveis. Essa estratégia explora a redundância natural das observações ao longo do tempo, reduzindo o impacto de erros pontuais de OCR e aumentando a robustez da identificação final.

A integração temporal das leituras contribui diretamente para a redução de falsos reconhecimentos e para uma associação mais confiável entre o veículo rastreado e sua identificação, especialmente em cenários urbanos reais, caracterizados por movimento contínuo e condições visuais adversas.

4.8 Integração do *SmarTTrack* à Plataforma *InterSCity*

A integração do modelo *SmarTTrack* à plataforma *InterSCity* constitui um elemento central da arquitetura proposta nesta dissertação, viabilizando a incorporação seletiva das informações produzidas pelo rastreamento veicular ao ecossistema urbano de dados. Diferentemente de abordagens voltadas ao monitoramento indiscriminado do tráfego, o modelo proposto restringe o registro e a publicação de dados exclusivamente a veículos de interesse das forças de segurança pública, tais como veículos furtados, roubados ou previamente monitorados.

O processo de integração ocorre de forma sistemática em duas etapas principais. Inicialmente, o dispositivo móvel realiza a sincronização periódica, em intervalos previamente definidos (por exemplo, a cada uma hora), de uma lista de veículos de interesse disponibilizada por meio de uma API mantida pelos órgãos de segurança pública. Essa lista contém exclusivamente os identificadores necessários à verificação local (e.g., placas veiculares) e é armazenada de forma temporária e controlada no dispositivo, em armazenamento privado da aplicação, com política de expiração e substituição a cada nova sincronização.

Após essa sincronização, cada sequência de caracteres de placa reconhecida durante a execução do sistema é confrontada localmente com a lista de veículos monitorados. Apenas quando a placa identificada corresponde a um registro presente nessa lista, a detecção é considerada válida para fins de registro do avistamento veicular na plataforma urbana. Detecções que não atendem a esse critério são descartadas no próprio dispositivo, não sendo encaminhadas à infraestrutura da *InterSCity*. Já no caso das detecções de interesse, são armazenadas na plataforma a placa, as coordenadas geográficas, a data e hora do avistamento e a imagem do veículo no momento da detecção.

Cabe destacar que o mecanismo de integração com a plataforma *InterSCity* é independente da estratégia de processamento adotada nas etapas anteriores do modelo.

Independentemente de o processamento dos vídeos ocorrer de forma local ou remota, a plataforma urbana consome apenas eventos consolidados de detecção e reconhecimento, não sendo impactada pela forma como o *pipeline* de visão computacional é executado.

Para os veículos de interesse, os dados associados à detecção são então registrados na plataforma *InterSCity*. Esse registro compreende o envio de eventos estruturados ao **Data Collector Service**, bem como a associação dos veículos detectados aos recursos correspondentes no **Resource Catalog**, utilizando identificadores padronizados. Cada evento armazenado inclui o conjunto mínimo de informações necessárias para análise e investigação, a saber: coordenadas geográficas (latitude e longitude), obtidas a partir dos sensores de localização do próprio dispositivo móvel mediante autorização do usuário; registro temporal da detecção (*timestamp*), fornecido pelo relógio do sistema operacional do dispositivo; imagem do veículo no momento da identificação; e a sequência de caracteres correspondente à placa veicular reconhecida.

Essa estratégia possibilita a reconstrução da rota percorrida por um veículo monitorado ao longo do tempo, a partir da correlação temporal e espacial das detecções registradas. Além disso, o envio dessas informações à plataforma permite a geração de alertas direcionados a agentes de segurança pública e a Centros de Comando e Controle estatais, favorecendo respostas rápidas e coordenadas em situações que demandam intervenção imediata.

De forma mais ampla, a integração entre o sensoriamento móvel participativo e a plataforma *InterSCity*, conforme implementada no *SmarTTrack*, promove uma abordagem colaborativa, escalável e orientada a eventos para o apoio às ações de segurança pública em Cidades Inteligentes. Ao limitar o armazenamento e a disseminação de dados apenas a veículos de interesse, o modelo contribui para a eficiência operacional, a redução do volume de dados processados e o uso responsável da infraestrutura urbana de informação.

4.9 Considerações Finais

Este capítulo apresentou a metodologia adotada para o desenvolvimento do modelo proposto de rastreamento veicular baseado em Sensoriamento Móvel Participativo, voltado ao contexto de Cidades Inteligentes. Foram descritas, de forma estruturada, as etapas que compõem o fluxo metodológico, desde a aquisição dos dados em cenários urbanos reais até a integração seletiva dos resultados em uma plataforma urbana de gerenciamento.

A metodologia contempla mecanismos de detecção, rastreamento e reconhecimento de placas veiculares, considerando tanto o processamento remoto quanto a execução local em dispositivos móveis, de modo a lidar com diferentes restrições computacionais e cenários de uso. Destacam-se a adoção de técnicas de visão computacional baseadas em aprendizado profundo, a integração temporal das leituras ao longo de múltiplos quadros e

a publicação restrita de eventos relacionados exclusivamente a veículos de interesse das forças de segurança pública.

Com isso, o capítulo estabelece o arcabouço metodológico necessário para a avaliação do modelo proposto, fornecendo base para a análise experimental apresentada no capítulo seguinte, na qual são investigados o desempenho, a robustez e as limitações da abordagem em diferentes cenários de execução.

5 Experimentos e Resultados

A condução dos experimentos apresentados neste capítulo tem como propósito avaliar a viabilidade prática do *SmarTTrack*, um modelo para rastreamento veicular baseado em sensoriamento móvel participativo voltado para aplicações em Cidades Inteligentes. Nesse modelo, cidadãos ou agentes públicos utilizam seus próprios dispositivos móveis para capturar vídeos e contribuir com informações de mobilidade urbana, permitindo, por exemplo, o monitoramento de veículos suspeitos ou o apoio investigativo em tempo real. Nesse contexto, compreender o comportamento do *pipeline* de visão computacional — incluindo detecção de veículos, rastreamento, extração de placas e OCR — torna-se essencial para estabelecer os limites operacionais da solução.

5.1 Objetivo dos Experimentos

O experimentos tiveram como objetivo analisar, de forma comparativa e sob condições controladas, três abordagens distintas para a execução do modelo *SmarTTrack* em dispositivos móveis: (i) o processamento integral realizado no próprio dispositivo; (2) o processamento remoto mediante envio contínuo do fluxo de vídeo; e (iii) o processamento remoto baseado no envio segmentado de blocos de vídeo. Essa comparação busca fornecer uma visão abrangente não apenas sobre o desempenho das métricas de detecção e rastreamento — como precisão, revocação e *F1-score* —, mas também sobre os custos computacionais associados a cada abordagem, incluindo consumo de CPU, uso de memória e volume de dados transmitidos pela rede móvel.

A análise estatística dos resultados obtidos, incluindo médias, desvios-padrão e intervalos de confiança, permite avaliar o comportamento e a consistência de cada cenário ao longo de múltiplas execuções. Dessa forma, torna-se possível identificar qual solução apresenta maior estabilidade operacional e qual modelo é mais adequado ao uso cotidiano em uma arquitetura de Sensoriamento Móvel Participativo, caracterizada pela heterogeneidade dos dispositivos e pela dependência de redes móveis potencialmente instáveis.

Ao integrar desempenho, eficiência e variabilidade, o experimento fornece evidências úteis para orientar a escolha do modelo de processamento mais viável às aplicações de rastreamento veicular em larga escala no contexto de Cidades Inteligentes.

5.2 Materiais

A execução local do sistema foi realizada em um *smartphone* **Samsung Galaxy A36 5G**, um dispositivo intermediário que representa adequadamente a realidade tecnológica de uma considerável parcela da população brasileira. Além de executar o cenário de processamento local, o mesmo aparelho foi utilizado para a captura dos vídeos e montagem do *dataset* empregado nos dois cenários de processamento remoto, garantindo uniformidade das condições experimentais. A opção por um dispositivo de desempenho médio leva em consideração que soluções de sensoriamento móvel participativo serão executadas por um público diversificado, em aparelhos com poder de processamento variável. Por isso a importância de se usar nos experimentos um hardware de capacidade média.

Para o processamento remoto, foi utilizada uma estação de trabalho de classe *workstation*, equipada com CPU *multicore* com GPU dedicada, simulando um ambiente de servidor de implantação em nuvem. O equipamento empregado foi um **Acer Predator Helios 300**, contendo um processador Intel Core i7 de 11^a geração, 16 GB de memória DDR4 e uma GPU NVIDIA RTX 3070 com 8 GB de VRAM. Essa configuração permite avaliar o comportamento do sistema em um cenário de maior disponibilidade computacional

5.3 Dataset

O experimento utilizou um vídeo com 5 minutos de duração, gravado a 30 quadros por segundo e resolução 1920×1080, em uma área de tráfego modernado na cidade de São Luís, Maranhão. A gravação foi realizada com o *smartphone* **Samsung Galaxy A36 5G**, e ocorreu no período da tarde, com iluminação natural, o que fornece condições realistas para avaliação da *pipeline*. Por conter carros, motocicletas e ônibus em diferentes velocidades, o vídeo é adequado para testar cenários completos de detecção, rastreamento e reconhecimento de placas.

5.4 Métricas

Para avaliar o custo computacional, foram utilizadas quatro métricas fundamentais. O tempo de CPU quantifica o esforço de processamento necessário para completar a execução do vídeo em cada cenário, refletindo a intensidade das operações realizadas. O delta de memória expressa a variação total de RAM utilizada pela aplicação durante o processamento, permitindo identificar picos de alocação e o impacto geral sobre os recursos do dispositivo. O volume de dados enviados mede a quantidade de tráfego gerado pela aplicação, sendo especialmente relevante nos cenários remotos, em que quadros ou blocos de vídeo precisam ser transmitidos pela rede móvel. Por fim, o consumo de bateria indica o

impacto energético de cada abordagem, permitindo avaliar sua viabilidade prática em um contexto de uso contínuo ou prolongado por cidadãos participantes. Como já mencionado, o *smarthphone* utilizado foi o **Samsung Galaxy A36 5G**, o qual sofreu um *reset*, sendo utilizado com as configurações-padrão de fábrica.

As métricas de desempenho seguem definições canônicas da visão computacional. A **Precisão** quantifica a proporção de detecções corretas entre todas as detecções realizadas; a **Revocação** mede a capacidade do sistema de identificar corretamente todos os objetos relevantes presentes na cena; e o **F1-score** combina as duas métricas por meio da média harmônica, equilibrando falsos positivos e falsos negativos quando há desequilíbrio entre eles.

Matematicamente, as métricas de desempenho utilizadas neste trabalho são definidas como:

$$Precision = \frac{TP}{TP + FP}, \quad (5.1)$$

onde TP (*True Positives*) representam as detecções corretas e FP (*False Positives*) correspondem às detecções incorretas.

$$Revocação = \frac{TP}{TP + FN}, \quad (5.2)$$

em que FN (*False Negatives*) representa os objetos presentes no vídeo que não foram detectados. A *Revocação* indica, portanto, a capacidade do sistema de não perder objetos relevantes.

Por fim, o *F1-score* é dado pela média harmônica entre *Precisão* e *Revocação*:

$$F_1 = 2 \cdot \frac{Precision \cdot Revocação}{Precision + Revocação}. \quad (5.3)$$

Essas métricas foram calculadas para cada execução do experimento e posteriormente analisadas estatisticamente.

5.5 Análise Estatística

A análise estatística baseia-se em três elementos fundamentais: média, desvio-padrão e intervalo de confiança. Cada cenário foi executado cinco vezes. O intervalo de confiança indica a faixa dentro da qual se espera que esteja o valor verdadeiro da média, considerando a variabilidade dos resultados observados. Neste trabalho, utilizou-se o intervalo de confiança de 95%, que expressa a incerteza associada às médias obtidas para cada métrica (*Precisão*, *Revocação* e *F1-score*). O cálculo foi realizado a partir da média

amostral e do desvio-padrão, considerando o número de execuções realizadas ($n = 5$). A fórmula geral do intervalo de confiança é:

$$IC = \bar{x} \pm t \cdot \frac{s}{\sqrt{n}}, \quad (5.4)$$

onde:

- $\bar{x}$ é a média amostral (por exemplo, a média das precisões obtidas nas cinco execuções);
- s é o desvio-padrão da amostra, que representa a variabilidade dos valores observados;
- n é o número de amostras, aqui igual a 5;
- t é o valor crítico da distribuição t de Student, calculado para nível de confiança de 95% com $n - 1$ graus de liberdade ($df = 4$).

Para essa configuração, o valor crítico é aproximadamente:

$$t \approx 2.776. \quad (5.5)$$

O termo

$$\frac{s}{\sqrt{n}}$$

corresponde ao erro padrão da média, que ajusta o desvio-padrão ao tamanho da amostra.

Assim, os limites do intervalo de confiança são dados por:

$$LI = \bar{x} - t \cdot \frac{s}{\sqrt{n}}, \quad (5.6)$$

$$LS = \bar{x} + t \cdot \frac{s}{\sqrt{n}}. \quad (5.7)$$

Esses valores definem o intervalo de confiança de 95%, isto é, a região onde existe uma probabilidade de 95% de que a média verdadeira do desempenho esteja contida. Na prática, intervalos mais estreitos representam resultados mais consistentes e maior estabilidade estatística, enquanto intervalos mais amplos indicam maior variabilidade entre as execuções e menor confiabilidade da média estimada.

5.6 Procedimentos e Métodos

A execução do experimento seguiu um procedimento padronizado. Inicialmente, o vídeo foi carregado no dispositivo móvel e analisado separadamente em cada um dos três cenários. Para garantir a confiabilidade dos resultados, cada cenário foi executado cinco vezes, de forma independente, permitindo posterior tratamento estatístico. Em cada execução, foram coletados os valores de no *smartphone*, a saber: consumo de CPU, variação consumo de memória, dados transmitidos e consumo de bateria. Foram calculadas ainda as métricas seguntes de desempenho do modelo: Precisão, Revocação e *F1-Score*. Todos os dados foram organizados em tabelas e submetidos à análise estatística baseada em média, desvio-padrão e intervalo de confiança de 95%.

5.7 Cenários Experimentais

O **Cenário 1** corresponde à execução integral da *pipeline* no próprio dispositivo móvel. Nesse caso, todo o processamento — desde a leitura dos quadros de vídeo até a identificação final da placa — ocorre localmente, sem a necessidade de conexão com servidor remoto.

No **Cenário 2**, o dispositivo móvel realiza a transmissão ao vivo de vídeo utilizando a biblioteca *RootEncoder*¹, que implementa o protocolo RTMP (*Real-Time Messaging Protocol*), amplamente empregado para o envio contínuo de áudio e vídeo pela internet. No início da transmissão, a câmera do dispositivo é acionada e o fluxo de vídeo é codificado em tempo real no formato H.264, operando a 30 quadros por segundo, o que garante um equilíbrio adequado entre qualidade visual e taxa de compressão.

Após a codificação, o vídeo é transmitido para um servidor de *streaming* responsável por atuar como intermediário entre o emissor e o sistema de processamento. Esse servidor recebe o fluxo RTMP dos dispositivos Android, realiza a decodificação e disponibiliza o vídeo em tempo real para as aplicações responsáveis pela análise. A transmissão ocorre por meio das redes móveis disponíveis — 5G, 4G ou 3G, conforme a cobertura local. O fluxo recebido é processado pelo sistema *SmarTTrack*, que executa o módulo de rastreamento de veículos sobre cada frame transmitido.

Já no **Cenário 3**, a aplicação realiza o envio de pequenos trechos de vídeo (*chunks*) para um servidor remoto responsável pelo processamento. A duração desses segmentos é um parâmetro configurável do sistema e, nos experimentos conduzidos nesta dissertação, foi definido o valor de 3 segundos. Essa escolha representa um compromisso entre a necessidade de fornecer ao servidor uma janela temporal suficiente para a realização confiável das etapas de detecção e rastreamento e a redução do tempo de resposta, do consumo de

¹ Biblioteca integrada ao ambiente Android responsável pela codificação e transmissão de fluxos de vídeo em tempo real, utilizada para encapsular e enviar quadros pela rede.

banda e da latência do processamento remoto. Considerando, por exemplo, uma taxa típica de captura de 30 quadros por segundo, cada segmento contém aproximadamente 90 quadros, quantidade adequada para a identificação de veículos e placas sem a sobrecarga associada ao envio de vídeos mais longos. Para organizar o fluxo de captura, montagem e transmissão desses segmentos, emprega-se uma arquitetura de produtor–consumidor, que permite o desacoplamento entre as etapas de extração de quadros e envio dos arquivos gerados.

O módulo produtor é responsável pela captura dos quadros, obtidos tanto a partir de vídeos previamente armazenados no dispositivo quanto da gravação em tempo real da câmera. À medida que são capturados, os quadros são inseridos sequencialmente em um *buffer* principal. O módulo consumidor atua sobre esse *buffer*, removendo os quadros na ordem de chegada e transferindo-os para um segundo *buffer*, cujo tamanho corresponde exatamente à quantidade de quadros necessária para formar um segmento de vídeo com a duração predefinida.

Quando o segundo *buffer* é completamente preenchido, inicia-se o processo de montagem do arquivo de vídeo, que é imediatamente preparado para envio ao servidor remoto. Durante a codificação e transmissão do segmento, o consumidor continua operando em paralelo, preenchendo novamente o segundo *buffer* conforme novos quadros chegam. Caso esse *buffer* atinja sua capacidade antes da conclusão do envio anterior, os quadros excedentes são descartados, evitando sobrecarga e garantindo continuidade ao fluxo de captura.

A transferência dos arquivos é realizada por meio do protocolo SFTP (*SSH File Transfer Protocol*), que opera sobre o canal criptografado do SSH (*Secure Shell*). Dessa forma, cada segmento de vídeo é transmitido com segurança, preservando a integridade dos dados e garantindo confidencialidade ponto a ponto. A conexão SSH é estabelecida assim que o processo de captura é iniciado, reduzindo o tempo de latência entre a geração e o envio dos segmentos subsequentes.

O envio dos arquivos ocorre pelas redes móveis disponíveis (5G, 4G ou 3G), variando conforme a qualidade da cobertura no momento da captura. Esse modelo introduz um comportamento distinto em relação ao *streaming* contínuo, uma vez que o tráfego passa a ocorrer de forma intermitente e orientada a blocos.

5.8 Resultados

A análise dos resultados obtidos neste estudo foi conduzida com base em dois eixos principais: (1) o desempenho do sistema, avaliado pelas métricas de Precisão, Revocação e *F1-score*; e (2) o consumo de recursos nos dispositivos móveis, considerando CPU, memória e dados enviados. Os valores numéricos obtidos em cada um dos três cenários

serão analisados a seguir. Cada conjunto de resultados é discutido à luz dos gráficos correspondentes, que permitem uma visualização mais clara das diferenças entre os três cenários avaliados.

5.8.1 Cenário 1 – Processamento Local no Dispositivo

Este cenário consiste na execução integral do *pipeline* de detecção e rastreamento diretamente no dispositivo móvel, sem dependência de infraestrutura remota. Dessa forma, todas as etapas — leitura de quadros, detecção de veículos, extração de placas e aplicação do OCR — são realizadas localmente, explorando exclusivamente os recursos de CPU e memória do smartphone. Esse modelo representa a configuração de máxima autonomia operacional, sendo particularmente relevante para aplicações de sensoriamento participativo em que a conectividade pode ser limitada ou intermitente.

No tocante ao desempenho, os resultados estatísticos obtidos a partir das cinco execuções independentes encontram-se apresentados na Tabela 2. Observa-se que o cenário local apresentou desempenho consistente, com métricas estáveis e baixa variabilidade entre as repetições. A Precisão média ficou em 60,12%, indicando que parte das detecções incorretas ainda ocorre, mas dentro de um comportamento previsível. A Revocação atingiu 81,56%, sendo a métrica mais elevada do cenário, demonstrando a capacidade do sistema de recuperar a maior parte das placas presentes nos vídeos. Já o *F1-score*, que sintetiza o equilíbrio entre Precisão e Revocação, obteve média de 69,06%, refletindo bom desempenho geral.

Os valores de desvio-padrão para as três métricas foram muito baixos (0,90 para Precisão; 0,33 para Revocação; e 0,76 para *F1-score*). Isso indica notável estabilidade estatística, evidenciando que as variações entre as execuções são discretas e que o modelo se comporta de maneira previsível. Os intervalos de confiança de 95% reforçam esse diagnóstico, pois permanecem estreitos e concentrados em torno das médias, revelando que o desempenho observado não é fruto de flutuação acidental, mas sim de um comportamento consistente do sistema.

Tabela 2 – Resultados de Desempenho para o Cenário 1(Local)

Métrica	Precisão	Revocação	F1-score
Média	60,12	81,56	69,06
desvio-padrão	0,90	0,33	0,76
95%	[59,00 – 60,98]	[81,15 – 81,89]	[68,11 – 70,00]

De forma geral, os resultados indicam que o processamento local apresenta boa estabilidade operacional, com destaque para a elevada revocação. No contexto da segurança pública, essa característica assume papel prioritário, uma vez que o principal objetivo é

maximizar a detecção de veículos de interesse na cena, reduzindo ao mínimo a possibilidade de omissões. Diferentemente de outros domínios, nos quais a precisão pode ser mais crítica, neste cenário a ocorrência de falsos positivos é aceitável, pois os resultados passam por uma etapa posterior de verificação por parte de agentes de segurança. Nesse sentido, a revocação de 81,56% evidencia a capacidade do sistema em identificar a maior parte dos veículos presentes, ampliando a cobertura de monitoramento. Já a precisão de 60,12%, embora indique a presença de detecções incorretas, não compromete a aplicabilidade prática, uma vez que cada evento pode ser acompanhado do envio de evidências visuais — como a imagem do veículo detectado — para a plataforma de Cidades Inteligentes, permitindo a validação humana antes de qualquer ação operacional. Assim, o *SmarTTrack* se mostra especialmente adequado como ferramenta de apoio à triagem e ao monitoramento inicial, priorizando a não omissão de alvos e contribuindo para o aumento da eficiência na identificação de veículos de interesse.

Observou-se, qualitativamente, que muitos erros de reconhecimento ocorreram por diferenças pontuais na sequência alfanumérica da placa, como a troca de um único caractere ou a identificação incorreta do tipo de placa, com consequente aplicação da máscara sintática inadequada. Como este trabalho adotou um critério estrito de acerto, considerando correta apenas a leitura integral dos sete caracteres, tais casos foram contabilizados como erro total. Esse resultado indica que parte das falhas corresponde a reconhecimentos parciais, aspecto que pode ser explorado em avaliações complementares futuras.

Com relação ao uso dos recursos do dispositivo, o cenário de processamento local demonstrou um comportamento igualmente consistente. O consumo médio de CPU foi de 41,00%, com Desvio-padrão reduzido (1,15), indicando baixa variabilidade entre as execuções. Esse resultado é compatível com a natureza do *pipeline*, que envolve decodificação do vídeo, execução contínua de modelos de visão computacional e operações de entrada/saída (e.g., geração de recortes e registros), o que impõe carga computacional sustentada ao longo do processamento. Ressalta-se que esse consumo é diretamente influenciado pelas capacidades de hardware e pelas condições de execução do dispositivo (e.g., CPU, disponibilidade de aceleração e carga do sistema). Ainda assim, a utilização observada permaneceu estável e dentro de limites suportados por um *smartphone* intermediário, como o utilizado nos experimentos (Samsung Galaxy A36 5G).

Quanto à memória, o consumo médio ficou em 248 MB, representando aproximadamente 3% da capacidade total do dispositivo (8 GB de RAM). Esse valor demonstra que, embora todo o *pipeline* seja executado localmente, o impacto sobre o sistema permanece baixo, permitindo que o dispositivo continue a operar normalmente mesmo com outros serviços ativos em segundo plano. A baixa variabilidade do consumo de memória reforça que o comportamento do sistema é estável e não apresenta flutuações inesperadas ao longo do processo. O consumo de bateria no cenário 1 foi em média 2%.

Por fim, como o processamento é integralmente embarcado, não há transmissão de dados ao servidor remoto durante a execução do experimento. Assim, o volume de dados enviados foi nulo, eliminando qualquer custo associado à conectividade móvel. Esse aspecto torna o modelo particularmente atraente para ambientes urbanos nos quais a largura de banda pode ser limitada, custosa ou sujeita a instabilidades.

Tabela 3 – Resultados de Consumo de Recursos para o Cenário 1 (Local)

Métrica	Consumo de CPU (%)	Delta de Memória (MB)
Média	41,00	248,00
desvio-padrão	1,15	1,41
IC 95%	[39,57 – 42,43]	[246,24 – 249,76]

5.8.2 Cenário 2 — Processamento Remoto via *Streaming*

O segundo cenário analisado corresponde ao processamento remoto do sistema, no qual o dispositivo móvel realiza apenas a captura contínua do vídeo e o transmite, em tempo real, para um servidor responsável por executar integralmente o *pipeline* de detecção, rastreamento e reconhecimento de veículos. Nesse modelo, o smartphone atua predominantemente como um emissor de dados, delegando ao servidor as operações computacionalmente intensivas. Essa mudança estrutural altera de maneira significativa o comportamento observado tanto no desempenho quanto no consumo de recursos quando comparado ao processamento local.

Os resultados de desempenho referentes a esse cenário são apresentados na Tabela 4. Observa-se que o processamento remoto via *streaming* apresentou média de Precisão ligeiramente superior à do processamento local, indicando uma redução no número de falsos positivos. Entretanto, essa vantagem não se estendeu às demais métricas. As médias de Revocação e *F1-score* foram inferiores às obtidas no cenário local e vieram acompanhadas de desvios-padrão mais elevados, evidenciando maior instabilidade entre as repetições. Os intervalos de confiança de 95%, mais amplos, reforçam que o comportamento do sistema é sensível às oscilações da rede móvel e às variações inerentes ao envio contínuo de quadros.

Tabela 4 – Resultados de Desempenho para o Cenário 2 (*Streaming*)

Métrica	Precisão	Revocação	F1-score
Média	61,66	77,04	68,39
desvio-padrão	2,43	4,83	1,78
Intervalo de confiança (IC 95%)	[58,65 – 64,68]	[71,04 – 83,03]	[66,18 – 70,60]

A análise estatística revela que, embora o servidor utilize maior poder computacional, a dependência da conectividade provoca perda eventual de quadros, flutuações na taxa

de transmissão e aumento significativo da variabilidade entre as execuções. Isso se reflete diretamente na redução da revocação, indicando que o sistema remoto deixa de identificar parte dos veículos presentes no vídeo, especialmente em condições de rede instável ou congestionada.

No que diz respeito ao consumo de recursos, o cenário de *streaming* apresenta comportamento inverso ao do processamento local. O uso de CPU e memória no dispositivo móvel é consideravelmente menor, uma vez que o aparelho não executa modelos de detecção ou reconhecimento. Contudo, essa economia vem acompanhada de um aumento expressivo no consumo de dados: cada execução do experimento resultou, em média, no envio de aproximadamente 175 MB de vídeo para o servidor, volume muito superior ao observado no cenário local, no qual não há transmissão de quadros.

Esse nível de utilização da rede possui implicações práticas relevantes, como possível vulnerabilidade à instabilidade da conexão, o que compromete sua viabilidade para uso cotidiano em arquiteturas de Sensoriamento Móvel Participativo. Em áreas com cobertura limitada, interferências ou alta densidade de usuários, o desempenho pode se degradar de forma significativa. O consumo de bateria no cenário 2 foi inferior a 1%.

Em síntese, o Cenário 2 demonstra que, embora o processamento remoto via *streaming* ofereça pequena vantagem em Precisão, ele apresenta fragilidades estruturais importantes: maior variabilidade estatística, redução na capacidade de detecção (revocação), dependência de conectividade e consumo elevado de dados. Esses aspectos tornam o modelo menos adequado para aplicações de rastreamento veicular em larga escala, especialmente em ambientes urbanos dinâmicos onde a estabilidade da rede não pode ser garantida.

Tabela 5 – Resultados de Consumo de Recursos para o Cenário 2 (*Streaming*).

Métrica	Consumo de CPU (%)	Delta de Memória (MB)
Média	14,99	77,30
Desvio-padrão	1,06	14,86
IC 95%	[13,67 – 16,31]	[58,85 – 95,76]

5.8.3 Cenário 3 — Processamento Remoto por Blocos de Vídeo

O terceiro cenário corresponde à abordagem de processamento remoto baseada no envio de blocos de vídeo, em que o dispositivo móvel grava pequenos segmentos, com duração de um a três segundos, e os envia ao servidor somente após a conclusão de cada bloco. Diferentemente do *streaming* contínuo, que exige transmissão ininterrupta dos quadros capturados, este modelo opera de maneira assíncrona e utiliza uma arquitetura de produtor-consumidor para gerenciar a captura, armazenamento temporário e envio dos lotes de quadros.

Do ponto de vista operacional, essa estratégia reduz a pressão de transmissão contínua sobre a rede móvel, mas introduz novos desafios, entre eles a possibilidade de descarte de quadros quando o buffer é preenchido mais rapidamente do que o servidor consegue processar. Além disso, a fragmentação dos segmentos pode limitar a continuidade temporal necessária para algumas etapas de rastreamento, influenciando diretamente as métricas de desempenho.

Os resultados obtidos nesse cenário são apresentados na Tabela 6. Nota-se que o processamento por blocos produziu uma média de Precisão superior aos outros dois cenários, indicando menor ocorrência de falsos positivos. No entanto, essa melhoria ocorreu em detrimento da Revocação, que registrou o menor valor médio entre os três modelos avaliados. Isso sugere que, embora as detecções corretas tenham sido mais precisas, parte dos veículos presentes nos vídeos não foi identificada, possivelmente devido à perda de continuidade temporal causada pelo envio segmentado.

O *F1-score*, que sintetiza o equilíbrio entre Precisão e Revocação, permaneceu em nível intermediário. Observa-se, contudo, um aumento expressivo nos desvios-padrão, revelando maior variabilidade entre as repetições. Essa dispersão é refletida também nos intervalos de confiança de 95%, substancialmente mais amplos do que os observados no processamento local, indicando que o comportamento do sistema é menos previsível nesse cenário.

Tabela 6 – Resultados de Desempenho para o cenário 3 (blocos de vídeo).

Métrica	Precisão	Revocação	F1-score
Média	65,65	71,11	68,06
desvio-padrão	5,11	6,63	3,93
IC 95%	[59,31 – 72,00]	[62,88 – 79,34]	[63,18 – 72,94]

Em relação ao consumo de recursos, o cenário de envio por blocos apresentou comportamento híbrido. O uso de CPU foi menor que no processamento integral no dispositivo, mas superior ao observado no envio contínuo de vídeo. Isso se deve ao custo de operações frequentes de compressão, geração e manipulação de arquivos temporários. O consumo de memória permaneceu em nível intermediário, compatível com o gerenciamento dos *buffers* necessários para agrupar os quadros antes do envio. Quanto ao tráfego de dados, o volume transmitido foi significativamente menor que no *streaming* contínuo, já que apenas segmentos compactados são enviados periodicamente, mas ainda maior que no processamento totalmente local. O consumo de bateria no cenário 2 se mostrou abaixo de 2%.

De maneira geral, este cenário apresentou resultados mistos. A maior Precisão sugere benefício em contextos onde a eliminação de falsos positivos é prioritária; entretanto,

a queda no valor médio da Revocação e o aumento da variabilidade estatística apontam para limitações importantes quando se exige cobertura abrangente das detecções, como em aplicações de segurança pública e rastreamento veicular. A combinação entre maior instabilidade, perda de continuidade temporal e dependência moderada da rede reduz a previsibilidade do sistema, tornando o processamento por blocos menos robusto que o processamento local e menos eficiente que o envio contínuo de vídeo quando se considera o conjunto das métricas.

Assim, o cenário 3 revela um equilíbrio intermediário entre custo e desempenho, mas não supera os demais modelos de forma consistente. Ainda que represente uma alternativa viável para situações com conectividade limitada, seus resultados estatísticos menos estáveis reforçam que essa abordagem deve ser empregada com cautela em sistemas de Sensoriamento Móvel Participativo que demandam alta confiabilidade operacional.

Tabela 7 – Resultados de Consumo de Recursos para o Cenário 3 (Blocos de vídeo)

Métrica	Consumo de CPU (%)	Delta de Memória (MB)
Média	25,17	143,51
Desvio-padrão	3,05	16,89
IC 95%	[21,38 – 28,95]	[122,53 – 164,48]

5.9 Comparação dos Cenários

Após a análise individual dos três cenários, esta seção apresenta uma síntese integrada dos resultados obtidos, destacando as diferenças estruturais entre as abordagens avaliadas. A comparação concentra-se nos intervalos de confiança de 95%, pois esses gráficos fornecem não apenas a estimativa pontual do desempenho e do consumo de recursos, mas também a medida de incerteza associada a cada estimativa. Assim, é possível identificar não apenas qual cenário apresenta os melhores valores médios, mas principalmente qual deles demonstra maior estabilidade estatística, fator essencial para aplicações reais de Sensoriamento Móvel Participativo em ambientes urbanos.

5.9.1 Desempenho do Sistema

A Figura 13 sintetiza os intervalos de confiança de 95% para Precisão, Revocação e *F1-score*. A partir dela, observa-se um contraste nítido entre os cenários. O processamento local apresenta intervalos visivelmente mais estreitos em todas as métricas, sugerindo um comportamento altamente estável entre as execuções. Esse padrão se destaca especialmente na revocação, que mantém valores elevados e com mínima dispersão, indicando forte capacidade de capturar a maioria dos veículos presentes no vídeo.

Em contraposição, os cenários remotos — tanto o envio contínuo quanto o envio por blocos — exibem intervalos mais amplos, refletindo maior variabilidade e dependência das condições de rede. O *streaming* contínuo apresenta redução considerável na revocação, enquanto o envio por blocos, apesar de elevar a precisão, amplia ainda mais a incerteza estatística, sobretudo devido às perdas de quadros decorrentes da segmentação.

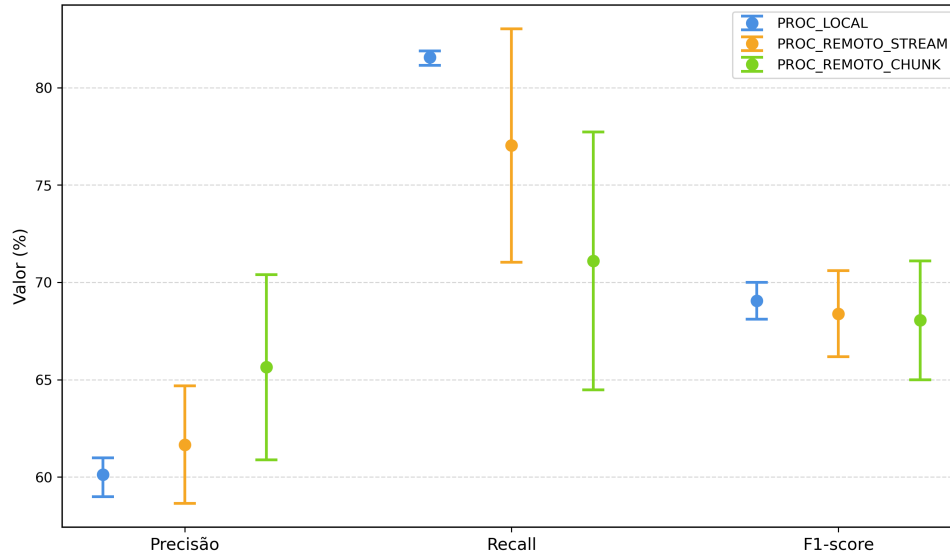


Figura 13 – Intervalo de confiança (IC 95%) das métricas de desempenho nos três cenários avaliados. Fonte: o autor

5.9.2 Consumo de CPU

A Figura 14 compara o intervalo de confiança do consumo de CPU. Embora o processamento local exija maior utilização média de CPU, seus intervalos estreitos revelam um comportamento previsível e controlado. Nos cenários remotos, em especial no *streaming* contínuo, a variabilidade aumenta sensivelmente, refletindo oscilações inerentes à captura, compressão e transmissão contínua de quadros. O envio por blocos mantém comportamento intermediário, porém ainda distante da estabilidade evidenciada pelo processamento local.

5.9.3 Consumo de Memória

A Figura 15 apresenta os intervalos de confiança para o uso de memória. Novamente, o comportamento do cenário local se destaca pela estabilidade: apesar de apresentar o maior consumo absoluto, sua variabilidade é mínima. Isso ocorre porque o *pipeline* completo de detecção e rastreamento permanece inteiramente alocado no dispositivo, sem depender de fatores externos.

Nos cenários remotos, ao contrário, a amplitude dos intervalos cresce significativamente. O envio contínuo apresenta flutuações substanciais devido à dinâmica

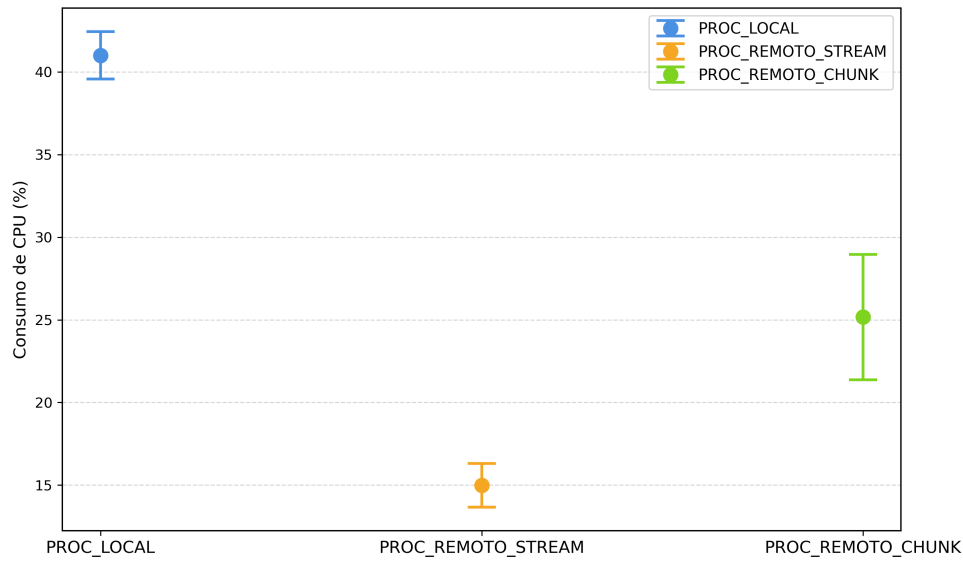


Figura 14 – Intervalo de confiança (IC 95%) do consumo de CPU nos três cenários. Fonte: o autor

da transmissão, enquanto o envio por blocos exibe comportamento ainda mais instável, fruto do processo descontínuo de captura, acúmulo e envio dos trechos de vídeo.

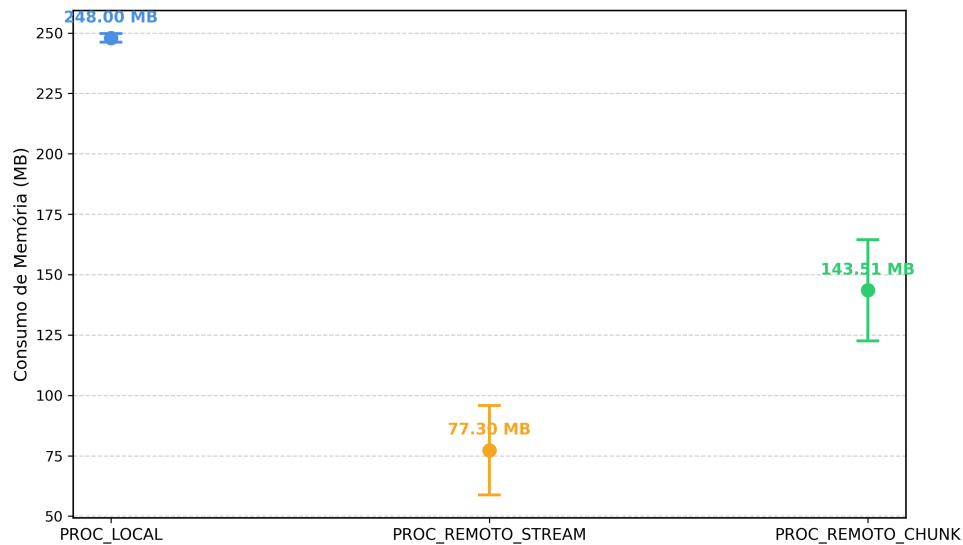


Figura 15 – Intervalo de confiança (IC 95%) do consumo de memória nos três cenários. Fonte: o autor

5.9.4 Consumo de Dados Móveis e Bateria

A Figura 16 sintetiza o volume de dados transmitidos nos três cenários avaliados. No cenário de processamento local, todo o *pipeline* é executado no próprio dispositivo móvel, não havendo transmissão de dados pela rede, o que elimina completamente a dependência de conectividade e quaisquer custos associados ao uso de banda móvel. Em

contraste, o cenário de *streaming* contínuo apresenta o maior consumo de dados, uma vez que todos os quadros capturados são enviados ao servidor remoto. O envio por blocos reduz esse volume ao transmitir apenas segmentos temporais do vídeo, porém mantém uma dependência significativa de rede e, conseqüentemente, de tráfego de dados.

Em relação ao consumo energético, os experimentos indicaram uma variação entre 1% e 2% de bateria nos três cenários, valor considerado aceitável para aplicações móveis em execução contínua. Esse nível de consumo é compatível com o uso prolongado em *smartphones* empregados em atividades profissionais, como no caso de motoristas de aplicativos, que frequentemente operam com múltiplas aplicações ativas durante longos períodos e, em muitos casos, mantêm o dispositivo conectado a uma fonte de alimentação veicular, mitigando o impacto prático dessa demanda energética adicional.

No que se refere ao impacto financeiro para o usuário, no cenário de processamento local — adotado como o mais adequado nesta dissertação — não há transmissão contínua de dados pela rede móvel, uma vez que o *pipeline* é executado integralmente no dispositivo. Assim, o uso de internet tende a se restringir a comunicações pontuais e de baixo volume (por exemplo, para verificação sob demanda e envio de eventos consolidados), tornando o custo associado a dados móveis residual. Essa característica reduz barreiras de adesão e favorece a aceitação do modelo de SMP pelos usuários.

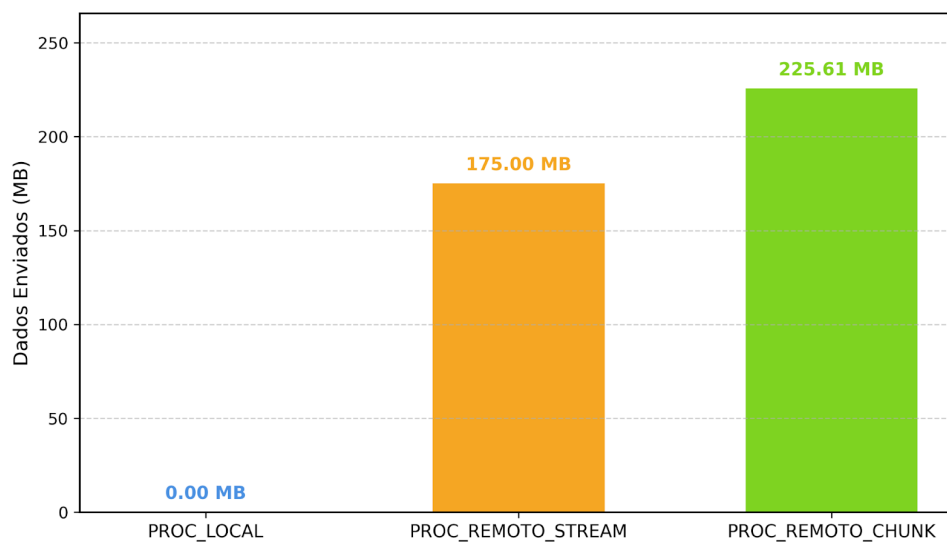


Figura 16 – Comparativo do consumo médio de dados móveis nos três cenários avaliados.
Fonte: o autor

5.9.5 Considerações Finais

A análise consolidada dos intervalos de confiança e das métricas médias permite comparar de forma objetiva os três cenários avaliados. Entre eles, o processamento local apresenta desempenho superior e mais estável. Seu valor médio de revocação (81,56%) é o

mais elevado entre os cenários, indicando maior capacidade de identificar corretamente os veículos presentes no vídeo. Consequentemente, o *F1-score* também se destaca (69,06%), resultado de um equilíbrio consistente entre precisão e cobertura. Além disso, os desvios-padrão e os intervalos de confiança são os menores entre as três abordagens, evidenciando uma execução estatisticamente mais previsível.

Em contraste, o envio contínuo de quadros apresenta uma queda significativa na revocação (77,04%) e um *F1-score* ligeiramente inferior (68,39%), acompanhado de intervalos de confiança mais amplos e maior variabilidade entre execuções. Já o envio por blocos, embora apresente a maior precisão média (65,65%), sofre forte redução na revocação (71,11%) e exibe o maior desvio-padrão entre os cenários, tornando-o o modelo menos estável estatisticamente.

No que se refere ao consumo de recursos, as diferenças são igualmente expressivas. O processamento local utiliza, em média, cerca de 247 MB de memória e mantém ocupados aproximadamente quatro núcleos de CPU durante a execução. Considera-se esses valores dentro dos limites aceitáveis para um *smartphone* intermediário. Por outro lado, o envio contínuo e o envio por blocos apresentam menor uso de CPU e memória, porém à custa de um consumo substancial de dados: o *streaming* contínuo demanda aproximadamente 175 MB por vídeo processado, enquanto o envio por blocos também impõe tráfego significativo, ainda que menor. O cenário local, ao contrário, elimina completamente o uso de rede para o processamento, resultando em maior autonomia energética e redução de custos operacionais.

Considerando simultaneamente o desempenho, a estabilidade estatística, a autonomia computacional e a viabilidade prática em ambientes urbanos heterogêneos, conclui-se que o processamento local é a abordagem mais adequada para aplicações de rastreamento veicular em arquiteturas de sensoriamento participativo móvel. Seu equilíbrio entre robustez, eficiência e independência de infraestrutura externa o torna a solução mais consistente para implantação em Cidades Inteligentes, com reais possibilidades de escalabilidade e adoção em larga escala.

6 Conclusões

Esta dissertação teve como objetivo investigar a viabilidade do uso do Sensoriamento Móvel Participativo aliado a técnicas de visão computacional para a detecção e rastreamento múltiplo de veículos no contexto de Cidades Inteligentes, com foco no apoio às ações de segurança pública. Partiu-se do desafio de ampliar a cobertura do monitoramento urbano sem dependência exclusiva de infraestrutura fixa, considerando restrições computacionais de dispositivos móveis, cenários de dupla mobilidade e a necessidade de uso responsável dos dados coletados.

Ao longo do trabalho, foram abordadas questões de pesquisa relacionadas à capacidade de dispositivos móveis executarem etapas de detecção, rastreamento e reconhecimento de placas veiculares de forma eficiente, bem como à adequação de diferentes estratégias de processamento (local e remoto) e à integração dos eventos gerados com uma plataforma urbana escalável. Os resultados obtidos permitem afirmar que tais objetivos foram alcançados, demonstrando a viabilidade técnica e arquitetural do modelo proposto.

O modelo desenvolvido mostrou-se capaz de integrar, de forma coerente, etapas de aquisição colaborativa de dados, detecção e rastreamento de veículos, reconhecimento automático de placas veiculares e publicação seletiva de eventos em uma plataforma urbana de gerenciamento. A avaliação experimental, conduzida em três cenários distintos — processamento local, *streaming* contínuo e envio por blocos — evidenciou que o processamento local apresenta maior estabilidade operacional, menor dependência de conectividade e consumo nulo de dados móveis, configurando-se como a estratégia mais adequada ao contexto de sensoriamento móvel participativo adotado.

Os resultados obtidos indicam que a combinação de redes neurais profundas para detecção e reconhecimento, associada à integração temporal das leituras ao longo de múltiplos quadros, permite identificar veículos de interesse das forças de segurança pública com elevado grau de confiabilidade, mesmo em cenários urbanos adversos. A distinção entre processamento local e remoto mostrou-se eficaz para lidar com diferentes restrições computacionais, enquanto a integração com a plataforma *InterSCity* viabilizou o registro estruturado de eventos e a reconstrução de trajetórias veiculares, apoiando ações de monitoramento e resposta operacional.

Este trabalho propôs e validou um modelo escalável de rastreamento veicular que amplia a cobertura do monitoramento urbano sem dependência exclusiva de câmeras fixas, explorando a participação cidadã de forma controlada, seletiva e orientada a eventos de interesse. Dessa forma, a pesquisa contribui para o avanço de soluções tecnológicas aplicadas à segurança pública e à mobilidade urbana, alinhando-se aos princípios de

eficiência, interoperabilidade e uso responsável de dados em Cidades Inteligentes.

6.1 Contribuições

As principais contribuições desta pesquisa são:

- Proposição de um modelo de rastreamento veicular de baixo custo baseado em Sensoriamento Móvel Participativo, voltado ao apoio às ações de segurança pública em Cidades Inteligentes, reduzindo a dependência de infraestruturas fixas de monitoramento.
- Desenvolvimento de uma arquitetura integrada para detecção, rastreamento e reconhecimento de placas veiculares, capaz de operar em cenários reais de dupla mobilidade utilizando dispositivos móveis amplamente disponíveis.
- Avaliação comparativa de estratégias de processamento local e remoto, evidenciando a viabilidade prática do processamento local em dispositivos móveis, com maior estabilidade de desempenho e menor dependência de conectividade.
- Integração do modelo proposto à plataforma de Cidades Inteligentes *InterSCity*, possibilitando o registro seletivo de eventos de interesse e a reconstrução de trajetórias veiculares para apoio à tomada de decisão por órgãos de segurança pública.

6.2 Limitações

Apesar dos resultados promissores, algumas limitações devem ser consideradas. Uma delas refere-se ao escopo da avaliação experimental, que foi conduzida em um conjunto controlado de cenários e dispositivos. Em particular, os experimentos foram realizados a partir de um conjunto de dados limitado, composto por um vídeo com duração de cinco minutos, o que restringe a variabilidade de situações observadas, como diferentes condições de tráfego, horários do dia e densidades urbanas. Embora esse volume tenha sido suficiente para validar a viabilidade técnica do modelo e comparar os cenários propostos, uma avaliação em larga escala, com maior diversidade de vídeos, usuários e ambientes urbanos, ainda se faz necessária.

Outra limitação relevante diz respeito ao desempenho quantitativo observado no cenário de processamento local, adotado como principal nesta dissertação. Conforme apresentado na Tabela 2, o modelo alcançou um *F1-score* médio de 69,06%, com precisão média de 60,12% e revocação de 81,56%. Embora a elevada revocação indique boa capacidade do sistema em identificar a maioria dos veículos de interesse, o valor de precisão mais reduzido evidencia a ocorrência de falsos positivos, impactando o desempenho global

do modelo nesse cenário. Ainda assim, os resultados demonstram um desempenho estável, com baixo desvio-padrão, indicando consistência nas execuções, e reforçam o caráter exploratório e incremental da proposta, cuja precisão pode ser ampliada em trabalhos futuros com bases de dados maiores, modelos mais robustos e estratégias adicionais de filtragem.

Observa-se, ainda, a ausência de comparação direta com soluções comerciais de mercado voltadas ao monitoramento veicular e ao reconhecimento automático de placas. Embora esse tipo de comparação seja relevante do ponto de vista aplicado, sua realização envolve desafios metodológicos importantes, como acesso restrito a sistemas proprietários, falta de transparência sobre os modelos utilizados e dificuldade de reproduzir, em condições equivalentes, os mesmos cenários experimentais adotados nesta dissertação. Assim, o foco deste trabalho concentrou-se na avaliação de diferentes arquiteturas de processamento a partir de um protótipo acadêmico funcional, desenvolvido especificamente para investigar a viabilidade do rastreamento veicular em cenários de dupla mobilidade e integração com plataformas de Cidades Inteligentes.

6.3 Trabalhos Futuros

Como trabalhos futuros, pretende-se ampliar e aprimorar o modelo proposto em diferentes direções. Uma possibilidade consiste no refinamento das etapas de reconhecimento óptico de caracteres, por meio do ajuste fino (*fine-tuning*) dos modelos de OCR com bases de dados específicas de placas veiculares brasileiras, contemplando variações regionais, condições adversas de iluminação, chuva e desgaste físico das placas.

Outra linha de investigação envolve a incorporação de atributos adicionais dos veículos ao processo de rastreamento, como cor, modelo e características visuais distintivas. A integração dessas informações pode aumentar a robustez do sistema em situações nas quais a leitura da placa é parcial ou temporariamente inviável, fortalecendo o suporte às atividades de segurança pública.

Do ponto de vista arquitetural, trabalhos futuros podem explorar a ampliação do modelo proposto para a integração de múltiplas fontes de dados, não se restringindo ao sensoriamento móvel participativo. Nesse contexto, o sistema pode incorporar fluxos de vídeo provenientes de câmeras externas autorizadas, como aquelas instaladas em residências, estabelecimentos comerciais ou pontos urbanos estratégicos, possibilitando a construção de uma arquitetura híbrida de computação de borda com processamento adaptativo conforme a conectividade e a capacidade computacional dos dispositivos envolvidos.

Adicionalmente, a ampliação da integração com a plataforma *InterSCity* pode contemplar a incorporação de novos tipos de eventos diretamente relacionados à segurança pública e à mobilidade urbana, como a detecção automática de acidentes de trânsito,

a identificação de situações anômalas envolvendo veículos e a correlação de múltiplas ocorrências veiculares em uma mesma região, favorecendo respostas mais rápidas e coordenadas por parte das autoridades competentes.

6.4 Publicações

Os resultados preliminares desta pesquisa foram publicados em evento científico internacional. Além disso, encontra-se em perspectiva a elaboração de versões estendidas do trabalho para futura submissão a periódicos científicos, contemplando a consolidação dos resultados e aprimoramentos desenvolvidos ao longo desta dissertação.

1. **SmarTTrack: A Mobile Crowdsensing System for Vehicle Multi-Tracking in Smart Cities.** NASCIMENTO NETO, Cícero Matias F. do; NUNES, Thales Gustavo Mendes; COUTINHO, Luciano Reis; SILVA, Francisco José da Silva e. In: *3rd International Conference on Edge Computing and Applications (ICECAA 2025)*. Namakkal, India, 3–4 Dec. 2025. Disponível em: <<https://icecaa.com>>. Situação: Artigo aceito e apresentado em sessão oral. DOI: Ainda não atribuído.

Tipo de publicação: Evento científico internacional.

Situação: Artigo aceito e apresentado em sessão oral online.

Qualis CAPES(2017-2020): A3 (Índice H5 = 22)

Referências

- ALMENDRA, P. N. V. *Pass-through da taxa de câmbio para a inflação no Brasil: Um estudo econométrico utilizando o filtro de Kalman*. Dissertação (Dissertação (Mestrado)) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015. Citado na página 31.
- ANANTHI, M. et al. Programmed number plate recognition using deep learning. In: *Proceedings of the 4th IEEE Global Conference on Advanced Technologies (GCAT)*. [S.l.: s.n.], 2023. Citado 6 vezes nas páginas 10, 29, 37, 49, 50 e 53.
- BAR-SHALOM, Y.; FORTMANN, T. E. *Tracking and Data Association*. Boston: Academic Press, 1988. Citado na página 33.
- BEWLEY, A.; GE, Z.; OTT, L.; RAMOS, F.; UPCROFT, B. Simple online and realtime tracking. *arXiv preprint arXiv:1602.00763*, 2016. Citado na página 32.
- BURKE, J.; ESTRIN, D.; HANSEN, M.; PARKER, A.; RAMANATHAN, N.; REDDY, S.; SRIVASTAVA, M. *Participatory Sensing*. Los Angeles, CA, 2006. Citado na página 22.
- CALITZ, A.; HILL, M. Automated license plate recognition using existing university infrastructure and different camera angles. *Journal of Information Systems*, v. 12, p. 195–217, 2020. Citado 3 vezes nas páginas 16, 36 e 37.
- CARDOSO, A. L. A. *Gerenciamento Descentralizado de Identidades para Cidades Inteligentes Baseado na Tecnologia Blockchain*. Dissertação (Mestrado) — Universidade Federal do Maranhão, 2024. Citado 2 vezes nas páginas 10 e 39.
- CHEN, G.-W. et al. Real-time license plate recognition and vehicle tracking system based on deep learning. In: *Proceedings of the 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*. [S.l.: s.n.], 2021. p. 378–381. Citado 2 vezes nas páginas 16 e 37.
- CHEN, H. et al. Crowdtracking: Real-time vehicle tracking through mobile crowdsensing. *IEEE Internet of Things Journal*, v. 6, n. 5, p. 7570–7583, 2019. Citado 9 vezes nas páginas 10, 15, 16, 25, 32, 33, 42, 43 e 53.
- CHRISTIN, D. Privacy in mobile participatory sensing: Current trends and future challenges. *Journal of Systems and Software*, v. 116, p. 57–68, 2016. Citado na página 24.
- DU, Y.; CUI, J.; LIU, B.; ZHANG, C.; LIN, Z.; LU, S.; WANG, Y.; YU, Y.; MA, J.; LING, H.; BAI, X. Pp-ocrv3: More attempts for real-world ocr. *arXiv preprint arXiv:2209.05425*, 2022. Citado na página 37.
- DU, Y. et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020. Citado na página 36.
- ESPOSTE, A. M. D.; KON, F.; COSTA, F. M.; LAGO, N. Interscity: A scalable microservice-based open source platform for smart cities. In: *Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems*. SCITEPRESS

- Science and Technology Publications, Lda, 2017. p. 35–46. ISBN 9789897582417. Disponível em: <<https://doi.org/10.5220/0006306200350046>>. Citado 2 vezes nas páginas 15 e 38.
- FBSP, F. B. de S. P. *17º Anuário Brasileiro de Segurança Pública*. 2023. FBSP. Dados referentes ao ano de 2022. Disponível em: <<https://forumseguranca.org.br>>. Citado 2 vezes nas páginas 16 e 17.
- FBSP, F. B. de S. P. *18º Anuário Brasileiro de Segurança Pública*. 2024. FBSP. Dados referentes ao ano de 2023. Disponível em: <<https://forumseguranca.org.br>>. Citado na página 16.
- FBSP, F. B. de S. P. *19º Anuário Brasileiro de Segurança Pública*. 2025. FBSP. Dados referentes ao ano de 2024. Disponível em: <<https://forumseguranca.org.br>>. Citado na página 17.
- FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, Elsevier, v. 55, n. 1, p. 119–139, 1997. Citado na página 48.
- GANTI, R. K.; YE, F.; LEI, H. Mobile crowdsensing: Current state and future challenges. *IEEE Communications Magazine*, v. 49, n. 11, p. 32–39, 2011. Citado 3 vezes nas páginas 10, 23 e 26.
- HERNÁNDEZ-MUÑOZ, J. M.; VERCHER, J. B.; MUÑOZ, L.; GALACHE, J. A.; PRESSER, M.; HÖLLER, L. A.; STILLER, T. Smart cities at the forefront of the future internet. In: *The Future Internet*. [S.l.]: Springer, 2011. p. 447–462. Citado 2 vezes nas páginas 20 e 22.
- JIANG, Z.; ZHU, H.; ZHOU, B.; LU, C.; SUN, M.; MA, X.; FAN, X.; WANG, C.; CHEN, L. Crowdpatrol: A mobile crowdsensing framework for traffic violation hotspot patrolling. *IEEE Transactions on Mobile Computing*, v. 22, n. 3, p. 1401–1416, 2023. Citado na página 25.
- JØRGENSEN, H. Automatic license plate recognition using deep learning techniques. In: . [s.n.], 2017. Disponível em: <<https://api.semanticscholar.org/CorpusID:55304067>>. Citado na página 35.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, v. 82, n. 1, p. 35–45, 1960. Citado 2 vezes nas páginas 33 e 61.
- KANTARCI, B.; MOUFTAH, H. T. Trustworthy sensing for public safety in cloud-centric internet of things. *IEEE Internet of Things Journal*, v. 1, n. 4, p. 360–368, 2014. Citado na página 25.
- KHAN, F.; Ur Rehman, A.; ZHENG, J.; JAN, M. A.; ALAM, M. Mobile crowdsensing: A survey on privacy-preservation, task management, assignment models, and incentives mechanisms. *Future Generation Computer Systems*, v. 100, p. 456–472, 2019. ISSN 0167-739X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167739X18315632>>. Citado na página 24.

- KIEW, L. C. et al. Vehicle route tracking system based on vehicle registration number recognition using template matching algorithm. In: *Proceedings of the International Conference on Software Engineering and Computer Systems (ICSECS) and the International Conference on Computer Science and Information Management (ICOCSIM)*. [S.l.: s.n.], 2021. p. 249–254. Citado 3 vezes nas páginas 33, 46 e 53.
- KON, F.; SANTANA, E. Cidades inteligentes: conceitos, plataformas e desafios. *Jornadas de Atualização em Informática - JAI*, p. 185–240, 2016. Citado 2 vezes nas páginas 21 e 22.
- KUSHWAHA, M. K.; SUSEELA, G. Car number plate detection using deep learning. In: *Proceedings of the International Conference on Data Science, Agents and Artificial Intelligence (ICDSAAI)*. [S.l.: s.n.], 2022. Citado na página 29.
- LANE, N. D.; MILUZZO, E.; LU, H.; PEEBLES, D.; CHOUDHURY, T.; CAMPBELL, A. T. A survey of mobile phone sensing. *IEEE Communications Magazine*, v. 48, n. 9, p. 140–150, 2010. Citado 2 vezes nas páginas 22 e 23.
- Laroca, R.; Severo, E.; Zanolrensi, L. A.; Oliveira, L. S.; Gonçalves, G. R.; Schwartz, W. R.; Menotti, D. A robust real-time automatic license plate recognition based on the YOLO detector. In: *International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018. p. 1–10. ISSN 2161-4407. Citado 2 vezes nas páginas 35 e 57.
- LIAO, M.; WAN, Z.; YAO, C.; CHEN, K.; BAI, X. *Real-time Scene Text Detection with Differentiable Binarization*. 2019. Disponível em: <<https://arxiv.org/abs/1911.08947>>. Citado na página 37.
- LUO, W.; XING, J.; MILAN, A.; ZHANG, X.; LIU, W.; KIM, T.-K. Multiple object tracking: A literature review. *Artificial Intelligence*, v. 293, p. 103448, 2021. ISSN 0004-3702. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0004370220301958>>. Citado na página 31.
- MUNKRES, J. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, v. 5, n. 1, p. 32–38, 1957. Citado na página 32.
- NEIROTTI, P.; MARCO, A. D.; CAGLIANO, A. M.; MANGANO, G.; SCORRANO, F. Current trends in smart city initiatives: Some stylised facts. *Cities*, v. 38, p. 25–36, 2014. Citado 2 vezes nas páginas 20 e 22.
- OHEKA, O.; TU, C. Real-time multiple vehicle detection using a rear camera mounted on a vehicle. In: *Proceedings of the International Conference on Intelligent and Innovative Computing Applications (ICONIC)*. [S.l.: s.n.], 2018. p. 1–5. Citado 6 vezes nas páginas 10, 32, 34, 47, 49 e 53.
- PICCARDI, M. Background subtraction techniques: a review. In: *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*. [S.l.: s.n.], 2004. Citado na página 32.
- PUARUNGROJ, W.; BOONSIRISUMPUN, N. Thai license plate recognition based on deep learning. *Procedia Computer Science*, v. 135, p. 280–287, 2018. Citado na página 36.

- RAHIMA, K.; HUSSAIN, M. Yolov11: An overview of the key architectural enhancements. *arXiv*, v. 2410.17725, 2024. Cs.CV. Disponível em: <<https://arxiv.org/abs/2410.17725>>. Citado na página 27.
- REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 779–788. Citado 4 vezes nas páginas 10, 26, 27 e 59.
- SANTANA, D. D. S. *Navegação terrestre usando unidade de medição inercial de baixo desempenho e fusão sensorial com filtro de Kalman adaptativo suavizado*. Tese (Tese (Doutorado)) — Universidade de São Paulo, São Paulo, 2011. Citado na página 31.
- SHASHIRANGANA, J.; PADMASIRI, H.; MEEDENIYA, D.; PERERA, C. Automated license plate recognition: A survey on methods and techniques. *IEEE Access*, v. 9, p. 11203–11225, 2021. Citado 3 vezes nas páginas 28, 35 e 37.
- SILVA, L. A. d. *Filtros de Kalman no tempo e frequência discretos combinados com subtração espectral*. Tese (Tese (Doutorado)) — Universidade de São Paulo, São Paulo, 2007. Citado 2 vezes nas páginas 31 e 32.
- SILVA, L. V. P. da. *Cidades Inteligentes e as Praças Centrais Urbanas: Uma Análise em São Luís (MA)*. Dissertação (Mestrado) — Universidade Federal do Maranhão, 2024. Citado na página 21.
- SPANU, M. et al. Smart cities mobility monitoring through automatic license plate recognition and vehicle discrimination. In: *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. [S.l.: s.n.], 2021. p. 1–6. Citado 3 vezes nas páginas 15, 44 e 53.
- TANG, Y.; XU, Y. C.; ZHANG, C. Z. Robust vehicle detection based on cascade classifier in traffic surveillance system. *The Open Automation and Control Systems Journal*, v. 6, p. 349–354, 2014. Citado na página 48.
- VIOLA, P.; JONES, M. Robust real-time face detection. *International Journal of Computer Vision*, Springer, v. 57, n. 2, p. 137–154, 2004. Citado na página 48.
- ZHENG, Y.; CAPRA, L.; WOLFSON, O.; YANG, H. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.*, Association for Computing Machinery, New York, NY, USA, v. 5, n. 3, set. 2014. Citado 2 vezes nas páginas 20 e 21.
- ZHU, L. et al. License plate recognition in urban road based on vehicle tracking and result integration. *Journal of Intelligent Systems*, v. 29, n. 1, p. 1587–1597, 2019. Citado 10 vezes nas páginas 10, 16, 29, 30, 32, 33, 37, 51, 52 e 53.