Monitoramento Inteligente de Tempo Real para Tráfego Urbano em Cidades Brasileiras

Carlos Loureiro, Elloá B. Guedes, Carlos M. S. Figueiredo

[1]Grupo de Pesquisa em Sistemas Inteligentes
Universidade do Estado do Amazonas
Av. Darcy Vargas, 1200 – Manaus – Amazonas
{cenlf.eng22, ebgcosta, cfigueiredo}@uea.edu.br

Abstract. *To contribute to real-time urban traffic monitoring in Brazilian cities, this work presents an intelligent solution utilizing images from urban surveillance cameras to train models for vehicle counting and classification. The YOLOv11 Medium model achieved an experimental mAP of* 0.7372 *at* 94 *FPS and was deployed in an interactive dashboard. This work also proposes a benchmark for the vehicle detection task and contributes to adaptive and responsive strategies for smart urban mobility in Brazilian cities.*

Resumo. *Visando contribuir para o monitoramento em tempo real do tráfego urbano em cidades brasileiras, este trabalho apresenta uma solução inteligente que utiliza imagens de câmeras de monitoramento urbanas para treinar modelos inteligentes de contagem e classificação de veículos. O modelo YOLOv11* Medium *obteve mAP experimental de* $0{,}7372$ *a* 94 *FPS, e foi implantado em um* dashboard *interativo. Este trabalho também propõe um* benchmark *para a tarefa de detecção de veículos e contribui com estratégias adaptativas e responsivas para a mobilidade urbana inteligente em cidades brasileiras.*

1. Introdução

O Brasil tem o transporte rodoviário como principal modal de deslocamento de cargas e passageiros ($65\,\%$), em que $78{,}1\,\%$ das rodovias estão sob a responsabilidade dos municípios [Moreira et al. 2018], nos quais habitam $87{,}4\,\%$ da população brasileira [IBGE 2022]. Estima-se que os brasileiros gastam, em média, 21 dias por ano no trânsito [Brasil 2022]. Mais recentemente, a mobilidade urbana no Brasil tem sofrido uma deterioração significativa, impulsionada pelo crescimento do transporte individual motorizado, a alta incidência de acidentes de trânsito, os congestionamentos crônicos e a poluição atmosférica causada por veículos automotores. A persistência desses problemas evidencia a necessidade urgente de políticas públicas que promovam a melhoria das condições de mobilidade e reduzam os custos dos deslocamentos urbanos [Carvalho 2016b]. Essas preocupações estão alinhadas ao Código de Trânsito Brasileiro (CTB), que incentiva soluções eficientes, seguras e sustentáveis para a mobilidade urbana [Brasil 1997].

Ao proporem a utilização de Tecnologias da Informação e Comunicação (TICs) para tornar os serviços urbanos mais eficientes, as Cidades Inteligentes (CIs) emergem como uma alternativa oportuna para o desenvolvimento e implantação de soluções sustentáveis e centradas nas necessidades dos cidadãos [Gassmann et al. 2019]. Neste contexto, a sinergia entre governos, tecnologia e participação cidadã apresenta-se como um caminho promissor para mitigar os desafios da mobilidade urbana [Savithramma et al. 2022], almejando contemplar melhorias no acesso, eficiência de locomoção, segurança e conforto para os usuários. Exemplos de soluções para mobilidade urbana em CIs no Brasil envolvem a detecção inteligente de

falhas em pavimentações asfálticas [Carvalho et al. 2022], a análise do impacto de bicicletas urbanas compartilhadas para favorecer outros modais de transporte [Cerutti et al. 2019], o uso de biocombustíveis no transporte público para reduzir a emissão de poluentes [Pasqual Lofhagen and Lira 2022], dentre outros.

Considerando a relevância da mobilidade urbana para o desenvolvimento urbano e a qualidade de vida da população [Carvalho 2016a], este trabalho propõe uma solução para impulsionar a consolidação das CIs no Brasil. Por meio da coleta de imagens de câmeras de trânsito em plataformas de dados abertos, modelos de Inteligência Artificial foram desenvolvidos para a detecção e classificação de veículos, viabilizando estimativas precisas e rápidas do fluxo de veículos. A integração da solução proposta com plataformas de gestão urbana possibilita a automação de processos, a análise de padrões de congestionamento, a otimização da temporização de semáforos, a inspeção automática de vias, a sugestão de modais de transporte aos cidadãos e o planejamento estratégico do posicionamento de agentes de trânsito em áreas de alto risco. A solução proposta colabora para otimizar recursos, reduzir o tempo de deslocamento e minimizar o impacto ambiental do tráfego urbano. Ao permitir a análise de grandes volumes de dados, viabiliza estratégias adaptativas e responsivas para a mobilidade urbana inteligente em cidades brasileiras.

Para apresentar o que se propõe, este trabalho está organizado em quatro seções principais: trabalhos relacionados (Seção 2), metodologia (Seção 3), resultados (Seção 4) e considerações finais com perspectivas futuras (Seção 5).

2. Trabalhos Relacionados

Para compreender de forma abrangente as metodologias das soluções existentes, identificar tendências e lacunas, realizou-se uma análise dos resultados de buscas no Google Acadêmico entre 2020 e 2024. Os critérios de inclusão contemplaram trabalhos baseados em Inteligência Artificial para detecção de veículos no Brasil.

Em relação à proposição de bases de dados para fomentar soluções inteligentes de monitoramento de tráfego, o trabalho de Campos *et al*. [2020] considerou a coleta de imagens oriundas de buscadores online, como o Google Street View, a utilização de imagens de outros autores e a modelagem gráfica para sintetizar imagens de sinalização de trânsito, que foram pouco frequentes nas buscas. O trabalho de Policarpo [2021] utilizou dados de vídeos de drones disponíveis no YouTube para a cidade de São Carlos (SP), considerando sete classes de veículos. Leal [2023], por sua vez, coletou vídeos de uma via urbana em Serra (ES) com o uso de um *smartphone* em posição fixa, obtendo também informações de velocidade. Observou-se, de maneira geral, que os *datasets* brasileiros são limitados, restringindo-se a uma uma ou poucas localidades, contemplando um único tipo de captura de dados e não tirando proveito da infraestrutura existente nas cidades.

No contexto de soluções para detecção de veículos em vias urbanas brasileiras, Santos *et al.* [2020] empregaram YOLOv3 e Deep SORT para detecção e rastreamento de veículos, respectivamente. A solução, avaliada em três datasets públicos, incluindo vídeos de cinco estradas federais, demonstrou precisão superior a $90\,\%$ após ajuste fino de hiperparâmetros. Considerada de baixo custo e não invasiva, a abordagem destaca-se pela eficácia em cenários experimentais. Ribeiro e Hirata [2023] também utilizaram modelos YOLO (YOLOv8 *Small*) associados a Ritmo Visual, uma técnica que gera imagens espaço-temporais a partir de vídeos. Os autores utilizaram quatro vídeos de domínio público obtidos a partir de câmeras estáticas em vias públicas, sem especificar a localização geográfica ou as características das vias. A solução proposta foi avaliada com mAP@$0{,}5$:$0{,}95$ igual a $0{,}691\,66$ e 186 FPS no conjunto

de teste. Os autores não mencionaram os custos de treinamento do modelo ou o número de parâmetros, mas indicaram oportunidades para melhorias.

Embora o monitoramento de tráfego urbano em tempo real seja um tópico relevante para o desenvolvimento de CIs, as contribuições na literatura que consideram a realidade brasileira ainda são escassas. A carência de bases de dados rotuladas para o problema no Brasil é uma das principais justificativas. As soluções existentes utilizam modelos YOLO (v3 e v8), mas há espaço para melhorias, como a exploração de versões mais recentes do modelo e a integração em um *pipeline* de monitoramento. As lacunas observadas, como a falta de dados e a necessidade de explorar modelos mais recentes, motivaram a realização do presente trabalho.

3. Materiais e Métodos

A metodologia deste trabalho consistiu de etapas para o desenvolvimento de um sistema inteligente de monitoramento de tráfego urbano, detalhado na Figura 1. Inicialmente, utilizou-se a infraestrutura de câmeras de monitoramento urbano e plataformas de dados abertos disponíveis em cidades brasileiras para construir uma base de dados rotulada, a partir da coleta e anotação de imagens de trânsito (Seção 3.1). Em seguida, foram implementados modelos de detecção de objetos (Seção 3.2), com avaliação de desempenho descrita na Seção 3.3, selecionando-se o modelo mais eficaz e eficiente. A solução foi então adaptada para uma cidade específica (Seção 3.4), possibilitando o monitoramento de vias urbanas e a previsão do quantitativo e tipo de veículos.

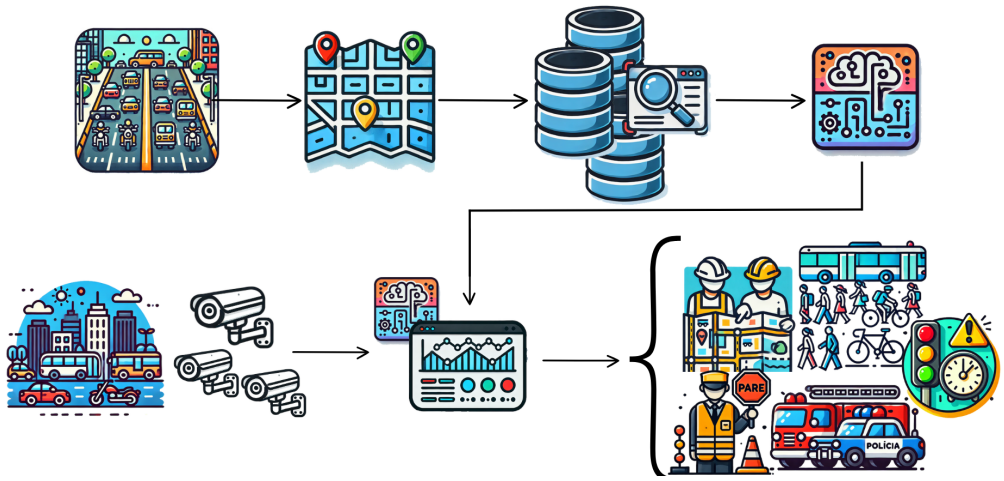


Figura 1: Visão geral da solução proposta.

3.1. Dados Experimentais: Aquisição, Pré-Processamento e Rotulação

A aquisição dos dados experimentais utilizados neste estudo foi realizada conforme ilustrado na Figura 2. Atualmente, diversas prefeituras divulgam vídeos em tempo real de câmeras de trânsito urbano na internet, por meio de sites próprios ou plataformas de *streaming*. Por meio de mecanismos de busca, foram selecionadas URLs desses serviços, considerando os critérios de disponibilidade, qualidade da imagem e variedade (de ângulos, iluminação, tipos de via, etc.), as quais foram armazenadas em um arquivo na nuvem. Para cada URL, foi elaborado um conjunto de parâmetros, que contemplam o nome da cidade associada, a posição do componente de vídeo na página, a taxa de amostragem para captura de imagens, dentre outros. Esses parâmetros foram agrupados em um arquivo de configuração JSON.

Desenvolveu-se então um *script* em Python para consultar o arquivo de URLs, recuperar os parâmetros de configuração associados e iniciar microserviços containerizados e paralelos de captura de tela, utilizando Docker e Selenium. Cada imagem capturada foi armazenada em um banco de dados estruturado, organizado por localidade, câmera associada e registro de data e hora. A solução proposta para a coleta de imagens é escalável, personalizável e flexível para integração com sistemas existentes.

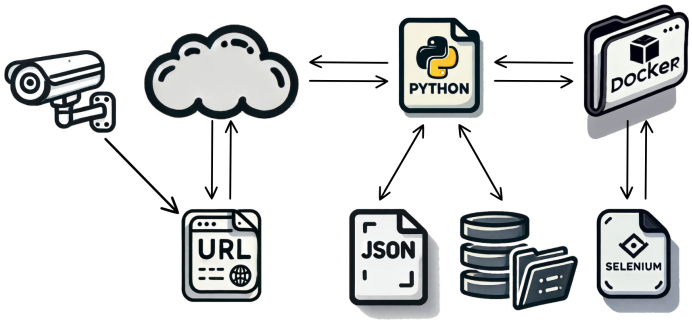
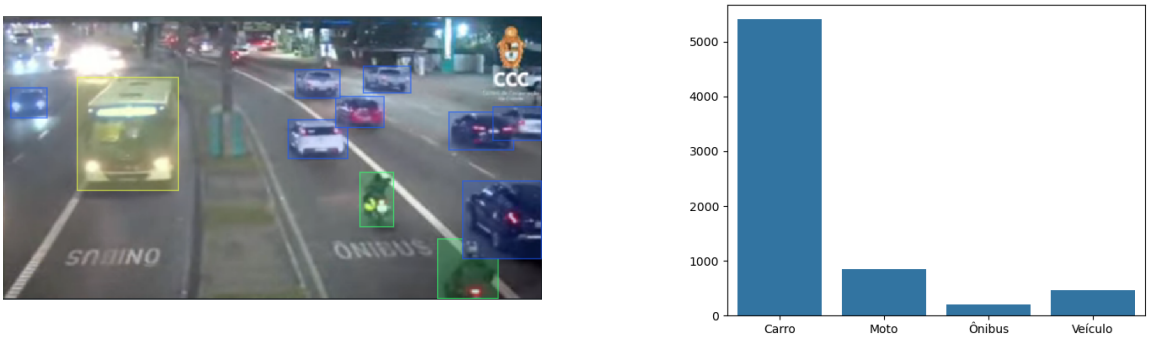


Figura 2: Diagrama esquemático da coleta de dados.

Com a utilização da solução de coleta de dados, foi elaborada uma base de dados contendo 1000 imagens no formato JPEG de vias urbanas brasileiras, provenientes de quatro cidades de diferentes regiões, sendo elas: Manaus (AM), Barueri (SP), Chapada (MT) e Porto Velho (RO). A base de dados possui 6932 anotações, divididas em quatro classes de objetos: carro, moto, ônibus e veículo (contemplando caminhões, vans, ambulâncias, etc.), característicos do trânsito urbano brasileiro. As anotações foram realizadas com a ferramenta Labelbox[1] por meio de caixas delimitadoras. A Figura 3 ilustra um exemplo de imagem rotulada e o histograma de distribuição de exemplos por classe, no qual se constata um evidente desbalanceamento.



(a) Exemplo Anotado **(b) Histograma dos exemplos**

Figura 3: Base de dados experimentais.

3.2. Modelos e Parametrização

Para a detecção de veículos nas imagens, foram avaliados os modelos listados a seguir, ambos baseados em *Deep Learning* e reconhecidos pela alta precisão e velocidade de processamento na detecção de objetos em tempo real.

[1]Fonte: `https://labelbox.com/`.

1. ***Region-Based Convolutional Neural Networks* (R-CNNs) da Família YOLO (*You Only Look Once*)**. Uma arquitetura de que reformula o problema da detecção de objetos como uma tarefa de regressão, realizando-o de maneira *single-shot* [Redmon et al. 2016]. Neste processo, o *backbone* atua como um extrator de características primário; o *neck* é responsável por agregar as características extraídas, combinando informações de diferentes escalas; o *head* produz as inferências; e um algoritmo de supressão não-máxima apriomora as detecções, removendo eventuais sobreposições. A técnica de *Mosaic Augmentation* atua na regularização do modelo, sintetizando imagens artificiais a partir dos exemplos disponíveis, colaborando para mitigar as dificuldades de detecção de objetos pequenos [Vijayakumar and Vairavasundaram 2024];
2. ***Real-Time Detection Transformer* (RT-DETR)**. Segundo uma arquitetura de *Transformers*, primeiramente extrai características da imagem de entrada com camadas convolucionais. Em seguida, um codificador processa as características em múltiplas escalas, um mecanismo de atenção aprende as partes relevantes da imagem, ponderando as diferentes partes da entrada, e um decodificador realiza as previsões [Carion et al. 2020]. Para mitigar os gargalos computacionais, o RT-DETR introduz um *backbone* baseado em convoluções e um codificador híbrido eficiente, baseado em convoluções e *Transformers* [Zhao et al. 2024].

Para a família YOLO considerou-se a versão YOLOv11 e os modelos *Nano*, *Small*, *Medium*, *Large* e *Extra-Large*. Num primeiro momento, todos esses modelos foram treinados com 500 épocas, paciência igual a 50 e demais hiperparâmetros em valores padrão. Em seguida, o ajuste fino dos modelos de melhor desempenho foi realizado de forma experimental, avaliando hiperparâmetros como tamanho de lote, taxa de aprendizado, técnicas de ajuste adaptativo da taxa de aprendizado e número de épocas. Para os modelos RT-DETR, foram consideradas as arquiteturas *Large* e *Extra-Large*, utilizando os parâmetros e hiperparâmetros padrão conforme a documentação do modelo. O treinamento foi realizado por até 500 épocas, com o objetivo de garantir a convergência dos modelos.

3.3. Avaliação de Desempenho

Para avaliação de desempenho, foi utilizada a técnica de validação cruzada *holdout*, com $60\,\%$ dos dados destinados ao treinamento, $10\,\%$ para validação e $30\,\%$ para testes. As métricas de desempenho dos modelos foram aferidas na partição de testes e contemplaram:

- **Precisão**. Proporção de veículos detectados corretamente dentre todas as detecções efetuadas;
- **Revocação**. Proporção de veículos reais corretamente identificados;
- **mAP@θ**. Avalia o desempenho de um modelo calculando a precisão média (AP) para cada classe de veículo (carro, moto, etc.) e, em seguida, obtendo a média dessas APs. O limiar θ define o quão sobrepostas duas caixas delimitadoras (a prevista e a real) precisam estar para que uma detecção seja considerada correta, fornecendo uma medida robusta da precisão do modelo na localização exata dos veículos [Padilla et al. 2020];
- ***Frames* por Segundo**. Refere-se à taxa de quadros de vídeo que o modelo de detecção é capaz de processar em um segundo.

3.4. Ambiente de Desenvolvimento e Implantação

O ambiente de desenvolvimento consistiu em um servidor computacional equipado com processador processador Intel® Core™ i7-8700 com clock de $3{,}2\,\mathrm{GHz}$, memória principal de $32\,\mathrm{GB}$, $960\,\mathrm{GB}$ de memória secundária e 2 placas de vídeo NVIDIA GTX 1080 Ti com $11\,\mathrm{GB}$ VRAM cada para aceleração em *hardware* do treinamento.

A solução foi implantada em Manaus (AM), capital brasileira com a 7ª maior população, com 2,27 Mi habitantes [IBGE 2022], e uma frota de aproximadamente 900 000 veículos, [Brasil 2022]. Adicionalmente à facilidade de aquisição de vídeos de monitoramento de trânsito no YouTube[2], um desafio prático do sistema viário da cidade representou um cenário adequado para avaliação realista da solução proposta: o sistema viário de Manaus possui 70 % de ocupação pelo transporte individual, o qual é utilizado por apenas 20 % dos habitantes[3], resultando em problemas como congestionamentos e ineficiência na mobilidade, poluição ambiental e aumento do risco de acidentes. Esse cenário reforça a demanda por soluções inteligentes para o contexto.

4. Resultados e Discussão

Os experimentos computacionais foram executados e aferiu-se o desempenho dos modelos de detecção de veículos na partição de testes da base de dados proposta. Os resultados dos modelos encontram-se dispostos na Tabela 1.

Tabela 1: Resultados dos modelos YOLOv11 para a primeira abordagem.

Modelo	Precisão	Revocação	F_1-*Score*	mAP@0,5	mAP@0,5:0,95	FPS	Épocas	Treinamento
YOLOv11 ***Nano***	80,21 %	80,53 %	80,37 %	0,8535	0,6467	172	119/500	705,50 s
YOLOv11 ***Small***	83,04 %	82,52 %	82,78 %	0,8774	0,7060	164	364/500	3312,18 s
YOLOv11 ***Medium***	85,26 %	80,71 %	82,92 %	0,8787	0,6942	93	125/500	2384,02 s
YOLOv11 ***Large***	82,96 %	79,41 %	81,15 %	0,8594	0,6754	74	114/500	2649,87 s
YOLOv11 ***Extra-Large***	87,22 %	79,81 %	83,35 %	0,8841	0,7147	46	240/500	9598,12 s

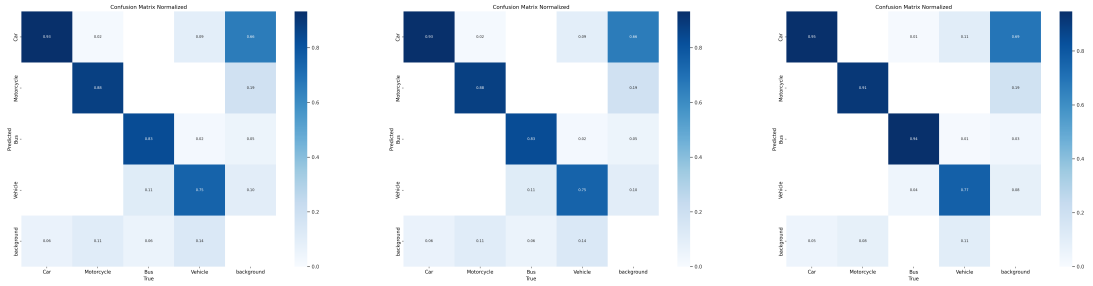
Utilizando o mAP@0,5:0,95, que avalia o desempenho em diferentes níveis de precisão de localização ao calcular a média do AP em limiares de IoU de 0,5 a 0,95, o modelo YOLOv11 *Extra-Large* obteve o melhor resultado. Apesar de maximizar essa métrica, observa-se que tal modelo possui a menor capacidade para detecção de tempo real (menor FPS) e o maior tempo de treinamento dentre os modelos avaliados. O YOLOv11 *Medium*, por exemplo, possui desempenho 2,85 % inferior a um custo computacional 4,17 vezes superior. O ganho de desempenho mostra-se especialmente discreto quando as matrizes de confusão na detecção dos objetos é considerada, conforme Figuras 4a e 4b, respectivamente. Em face desse cenário, oportunizou-se o ajuste fino dos modelos, o qual considerou 56 parametrizações diferentes, cujos 5 melhores resultados são mostrados na Tabela 2, em que os hiperparâmetros detalhados são o *batch size* e o *momentum*. Em todos esses ajustes, o melhor otimizador foi o SGD, foram utilizadas 600 ou 800 épocas e a taxa de aprendizado inicial foi igual a 0.01.

Tabela 2: Desempenho dos 5 melhores modelos YOLO após ajuste fino dos hiperparâmetros.

Modelo	Precisão	Revocação	F_1-*Score*	mAP@0,5	mAP@0,5:0,95	FPS	Épocas	Treinamento	Hiperparâmetros
YOLOv11 ***Medium***	87,03 %	84,55 %	85,77 %	0,9120	0,7372	94	245/600	6297,09 s	(8, 0.9)
YOLOv11 ***Large***	88,97 %	82,14 %	85,42 %	0,9038	0,7448	74	249/800	5887,72 s	(16, 0.7)
YOLOv11 ***Medium***	81,70 %	85,52 %	83,57 %	0,9030	0,7322	94	444/600	11 377,90 s	(8, 0.95)
YOLOv11 ***Medium***	84,42 %	85,95 %	85,18 %	0,9020	0,7313	94	200/600	5134,29 s	(8, 0.8)
YOLOv11 ***Medium***	83,42 %	86,63 %	85,00 %	0,9017	0,7320	94	279/600	5536,54 s	(16, 0.9)

A partir de uma maximização do mAP@0,5:0,95 e do FPS, percebe-se que o ajuste fino foi eficaz para melhorar o modelo YOLOv11 *Medium*, o qual superou a YOLOv11 *Extra-*

[2]Fonte: `https://tinyurl.com/5bfemhcb`.
[3]Fonte: `https://tinyurl.com/3yu636mj`.



(a) YOLOv11 *Extra Large* **(b) YOLOv11 *Medium* Tradicional** **(c) YOLOv11 *Medium* com *fine-tunning***

Figura 4: Comparação das matrizes de confusão entre modelos avaliados.

Large original em $3{,}25\,\%$ com FPS $2{,}04$ maior e tempo de treino $1{,}52$ vezes menor, com matriz de confusão detalhada na Figura 4c. Menciona-se também que o modelo YOLOv11 *Medium* possui $20{,}1$ Mi parâmetros, enquanto o modelo *Extra-Large* possui $56{,}9$ Mi, o que reforça o decréscimo no ônus computacional aliado ao ganho de desempenho na tarefa. O modelo YOLOv11 *Large* obteve aumento no mAP@$0{,}5$:$0{,}95$, entretanto a taxa de detecção sofreu um decréscimo significativo, impactando sua aplicabilidade em cenários práticos.

O desempenho dos modelos RT-DETR é apresentado na Tabela 3. Observa-se que os resultados não superaram os dos modelos YOLO avaliados após o ajuste fino. As justificativas para esse desempenho são atribuídas ao pequeno volume de dados disponível para treinamento e às dificuldades inerentes à otimização e depuração de modelos baseados em atenção.

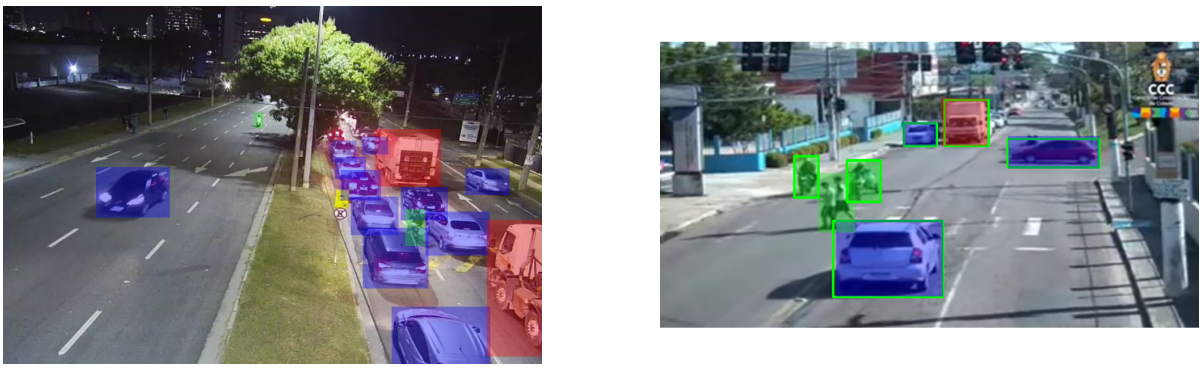
Tabela 3: Resultados dos modelos RT-DETR para a tarefa considerada.

Modelo	Precisão	Revocação	F_1-*Score*	mAP@$0{,}5$	mAP@$0{,}5$:$0{,}95$	FPS	Épocas	Treinamento
RT-DETR *Large*	$85{,}48\,\%$	$79{,}55\,\%$	$82{,}41\,\%$	$0{,}8596$	$0{,}6876$	29	$72/500$	$2096{,}49\,\text{s}$
RT-DETR *Extra-Large*	$83{,}99\,\%$	$86{,}67\,\%$	$85{,}31\,\%$	$0{,}8787$	$0{,}7070$	20	$74/500$	$3378{,}31\,\text{s}$

Considerando o modelo YOLOv11 *Medium* após ajuste fino como o mais adequado ao problema em questão, a Figura 5 ilustra exemplos de detecção de veículos no conjunto de testes. Na Figura 5a, observa-se uma detecção de alta qualidade, em que todos os veículos de ambas as vias foram corretamente delimitados e classificados em uma situação noturna. A Figura 5b ilustra um erro recorrente do modelo, o qual rotulou pedestres como motocicletas. Esse erro pode ter sido influenciado pela ausência de objetos do tipo pedestre no treino, fazendo com que o modelo os classifique como motocicleta pela maior semelhança.

A solução foi integrada a um protótipo de *dashboard* para visualização das informações. O monitoramento realizou-se em quatro vias urbanas de Manaus (AM) no dia 06 de março de 2024, com contagem e classificação de veículos a cada dez minutos. O *dashboard* interativo é apresentado na Figura 6, em que a parametrização da situação da via foi personalizada com base na média histórica de veículos no mesmo horário. A restrição de circulação seguiu o Decreto Municipal Nº 2.100/2013, que estabelece normas para o tráfego de veículos pesados na cidade.

Uma semelhança da solução proposta com os trabalhos relacionados reside no tipo de modelo utilizado, em que a família YOLO demonstrou eficácia e eficiência. Particularmente, há diferenças nos cenários experimentais, principalmente oriundos dos diferentes *datasets* uti-



(a) (b)

Figura 5: Exemplos de detecção realizadas pela solução de referência.

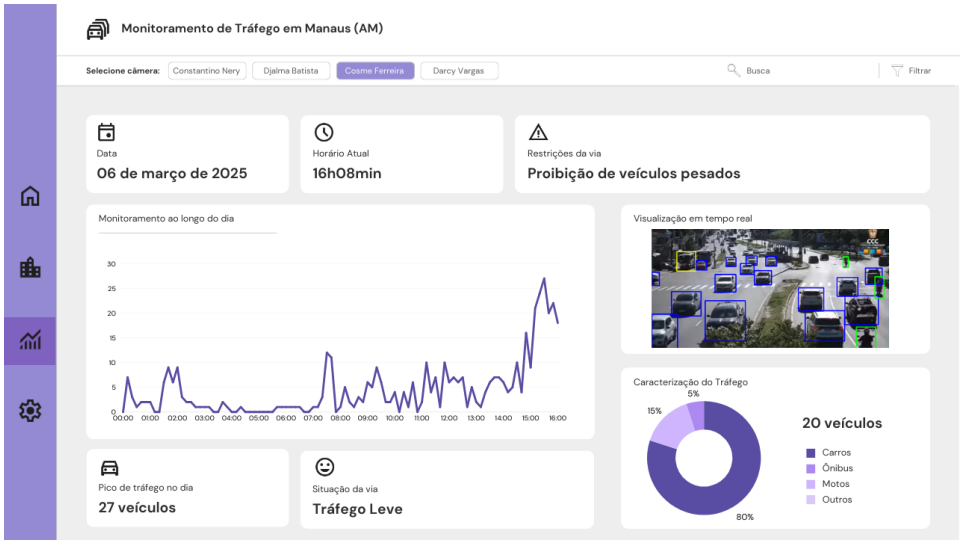


Figura 6: *Dashboard* interativo de monitoramento.

lizados. Logo, não é possível uma comparação mais objetiva entre eles. No entanto, este trabalho considera a integração da solução em uma arquitetura mais completa, com coleta de dados em condições reais e aplicação dos modelos para a criação de visualizações em tempo real, aproximando a contribuição a cenários reais de Cidades Inteligentes.

5. Considerações Finais

Visando contribuir para o monitoramento inteligente em tempo real do tráfego urbano em cidades brasileiras, o presente trabalho propôs uma solução flexível que contempla desde a captura de imagens de trânsito urbano disponíveis de câmeras de monitoramento *online* até a criação de uma base de dados com 1000 imagens e 6932 anotações, provenientes de diversas cidades brasileiras, para treinamento e teste de modelos inteligentes. A base de dados foi utilizada para treinar um modelo YOLOv11 *Medium* com ajuste fino de hiperparâmetros, que obteve mAP@0,5:0,95 de 0,7372 no cenário experimental proposto, efetuando detecções a 94 FPS, demonstrando eficiência na detecção em tempo real. Por conseguinte, o modelo foi integrado a um *dashboard* interativo, que exibe dados de contagem e classificação de veículos, demonstrando a viabilidade da implantação em um cenário real.

Uma das contribuições deste trabalho é o *dataset* elaborado, disponível em `doi.org/`

10.17632/m45vcmndvc.1, que pode ser utilizado como *benchmark* para soluções similares na literatura, favorecendo o desenvolvimento de soluções inteligentes para as demandas nacionais. O protótipo proposto, disponível em https://github.com/Idkode/sbcup-2025/, compreendendo a solução de captura de imagens, pode ser expandido para mais cidades brasileiras e personalizado para cenários individuais, que exijam soluções diferenciadas. O modelo proposto, também disponível no mesmo repositório, pode ser utilizado diretamente ou mediante transferência de aprendizado em outras localidades.

Como trabalhos futuros, pretende-se expandir a avaliação da solução por meio da captura de mais câmeras urbanas e da inclusão de mais objetos de interesse, como pedestres e ciclistas. Além disso, pretende-se avaliar o potencial de uso da solução no contexto do Plano Nacional de Contagem de Tráfego do Departamento Nacional de Infraestrutura de Transportes (DNIT), analisando sua precisão e eficiência em cenários reais. Finalmente, pretende-se disponibilizar o *dashboard* interativo por meio de uma plataforma web, para facilitar o acesso e a utilização da solução por cidadãos, pesquisadores e governos.

Transparência da IA e Considerações Éticas

Os ícones apresentados nas Figuras 1 e 2 foram gerados utilizando o modelo DALL-E 3. A revisão gramatical e ortográfica foi realizada com a ferramenta Gemini 2.0 Flash. Além disso, todas as revisões automáticas foram complementadas por uma revisão humana para garantir a precisão e a qualidade do texto.

Agradecimentos

Os autores agradecem o apoio financeiro concedido pela Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) por meio do programa PAIC 2024/2025. EBG agradece o apoio financeiro da CAPES por meio do Projeto COFECUB-PJ3126170P. Os autores agradecem também o apoio material recebido do Laboratório de Sistemas Inteligentes (LSI) da Universidade do Estado do Amazonas (UEA).

Referências

Brasil (1997). Lei Nº 9.503, de 23 de setembro de 1997 – Código de Trânsito Brasileiro. Disponível em https://tinyurl.com/3pejezb4. Acesso: 03/mar/2025.

Brasil (2022). Ministério das Cidades – Pesquisa Nacional de Mobilidade Urbana (PEMOB) 2022. Technical report. Disponível em: https://tinyurl.com/z4mhvy67. Acesso: 10/mar/2025.

Campos, D. H. C., Rodrigues, E. d. O., and Campos, E. C. (2020). Criação e validação de uma base de dados com elementos do trânsito brasileiro para veículos autônomos. *Brazilian Applied Science Review*, 4(3):1578–1590.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer.

Carvalho, C. H. d. (2016a). Mobilidade urbana: avanços, desafios e perspectivas. In Costa, M. A., editor, *O Estatuto da Cidade e a Habitat III: um balanço de quinze anos da política urbana no Brasil e a nova agenda urbana*, pages 345–364, Brasília. IPEA.

Carvalho, C. H. R. (2016b). Desafios da mobilidade urbana no Brasil. Technical Report 2198, Brasília. Disponível em https://tinyurl.com/25ezdjj8. Acesso: 03/mar/2025.

Carvalho, R., Guedes, E., and Figueiredo, C. (2022). Detecção Inteligente de Falhas em Pavimentações Asfálticas com Redes Neurais Convolucionais Regionais. In *Anais do III Workshop Brasileiro de Cidades Inteligentes*, pages 119–130, Porto Alegre, RS, Brasil. SBC.

Cerutti, P. S., Martins, R. D., Macke, J., and Sarate, J. A. R. (2019). "Green, but not as green as that": An analysis of a Brazilian bike-sharing system. *Journal of Cleaner Production*, 217:185–193.

Gassmann, O., Böhm, J., and Palmié, M. (2019). *Smart Cities: Introducing Digital Innovation to Cities*. Emerald Publishing Limited.

IBGE (2022). Instituto Brasileiro de Geografia e Estatística – Censo Demográfico 2022. Disponível em `https://tinyurl.com/yc2vyj7d`. Acesso: 03/mar/2025.

Leal, D. M. (2023). Detecção e Rastreamento de Objetos em Vídeo via Rede Neural Convolucional (CNN). Instituto Federal do Espírito Santo. Curso de Engenharia de Controle e Automação (Trabalho de Conclusão de Curso).

Moreira, M. A. L., Freitas Junior, M. d., and Toloi, R. C. (2018). O transporte rodoviário no Brasil e suas deficiências. *Refas - Revista FATEC Zona Sul*, 4(4):1–13.

Padilla, R., Netto, S. L., and da Silva, E. A. B. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, Niterói, Brasil.

Pasqual Lofhagen, J. C. and Lira, G. S. d. (2022). Cidades Inteligentes e o Transporte Urbano Sustentável com Bioenergia: Um Estudo de Caso de Curitiba, Brasil. *Rev. Tecnol. Soc.*, 18(51):207.

Policarpo, V. A. (2021). Análise de datasets para a detecção de veículos em imagens aéreas com algoritmos de Deep Learning e aplicação para a cidade de São Carlos. Escola de Engenharia de São Carlos, Universidade de São Paulo (Trabalho de Conclusão de Curso).

Redmon, J., Divvala, S., Girshick, R. B., and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Ribeiro, V. and Hirata, N. (2023). Combining YOLO and Visual Rhythm for vehicle counting. In *Anais Estendidos da XXXVI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 164–167. SBC.

Santos, A. M., Bastos-Filho, C. J. A., Maciel, A. M. A., and Lima, E. (2020). Counting vehicle with high-precision in Brazilian roads using YOLOv3 and Deep SORT. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE.

Savithramma, R. M., Ashwini, B. P., and Sumathi, R. (2022). Smart Mobility Implementation in Smart Cities: A Comprehensive Review On State-of-art Technologies. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 10–17, India. IEEE.

Vijayakumar, A. and Vairavasundaram, S. (2024). YOLO-based object detection models: A review and its applications. *Multimed. Tools Appl.*

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., and Chen, J. (2024). DETRs beat YOLOs on real-time object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16965–16974. IEEE.