

Detecção de Semáforos em Ambientes Urbanos para Cidades Inteligentes

Paulo Sérgio Da Silva Freitas Júnior* Elloá B. Guedes*

* Grupo de Pesquisa em Sistemas Inteligentes
Universidade do Estado do Amazonas (UEA)
Av. Darcy Vargas, 1200 – Manaus – Amazonas
(e-mail: {psdsfj.snf20,ebgcosta}@uea.edu.br)

Abstract: This study encompasses the recognition of urban traffic lights by conducting a comparative performance analysis of YOLOv5, YOLOv7, and YOLOv8 R-CNN models. Two datasets were used for this purpose: the Bosch Small Traffic Lights Dataset and BOSCHv2. The former consists of 13,425 images and 24,000 annotations related to traffic lights captured in the areas of Francisco Bay Area and Palo Alto, California, while the latter is a derivative containing annotations exclusively for medium and large-sized traffic lights. The experiments conducted with holdout cross-validation revealed that the YOLOv8 Nano and YOLOv8 Small models achieved the best performances on the BOSCHv2 dataset, with mAP@0.5 metrics of 0.826 and 0.851, respectively. These findings highlight the models' strong performance when applied to the proposed data preparation and contribute to the development of solutions for traffic light detection in Smart Cities.

Resumo: Este trabalho contempla o reconhecimento de semáforos urbanos realizando uma análise comparativa de desempenho dos modelos das famílias YOLOv5, YOLOv7 e YOLOv8. Para tal, foram utilizados dois conjuntos de dados: Bosch Small Traffic Lights Dataset e BOSCHv2, sendo o primeiro composto por 13.425 imagens e 24.000 anotações referentes aos semáforos captados nas áreas de Francisco Bay Area e Palo Alto, Califórnia e o segundo sendo um derivativo com apenas anotações de semáforos consideradas médias e grandes. Os experimentos realizados com validação cruzada holdout evidenciaram que os modelos YOLOv8 Nano e YOLOv8 Small no dataset BOSCHv2 obtiveram os melhores desempenhos, com a métrica mAP@0.5 de 0.826 e 0.851, respectivamente. As descobertas encontradas ressaltam um bom desempenho dos modelos frente à preparação de dados proposta e colaboram para o desenvolvimento de soluções para detecção de semáforos em Cidades Inteligentes.

Keywords: Deep Learning; Object Detection; YOLO; Smart Cities; Traffic Lights.

Palavras-chaves: Aprendizado Profundo; Detecção de Objetos; YOLO; Cidades Inteligentes; Semáforos.

1. INTRODUÇÃO

Cidades Inteligentes (CIs) são caracterizadas pelo uso estratégico, sistemático e coordenado de Tecnologias da Informação e Comunicação em uma variedade de contextos urbanos, objetivando satisfazer de forma abrangente a necessidade dos seus cidadãos alcançando a sustentabilidade econômica, social e ambiental (Song et al., 2017). Sob diversas perspectivas, uma CI se propõe a otimizar o uso dos recursos e infraestrutura de uma forma sustentável com vistas a melhorar a qualidade de vida da população que nela habita (Kon and Santana, 2016).

Giffinger *et al.* definem seis dimensões a serem consideradas para mensurar o grau de desenvolvimento de uma CI, as quais consideram economia (empresas instaladas, ambiente para empreendedorismo, etc.), população (educação, emprego, renda, etc.), governança (transparência de órgãos públicos, facilidade no uso de serviços públicos, etc.), meio-ambiente (poluição ambiental, eficiência no uso da água, etc.), vida inteligente (entretenimento, segurança,

áreas verdes, bibliotecas, etc.) e mobilidade. No tocante à mobilidade, em especial, menciona-se a facilidade de usar diferentes modais de transporte (bicicleta, carro, ônibus, metrô, etc.), quantidade e tamanho de congestionamentos e aplicações para facilitar e incentivar o uso de transporte público e sustentável, dentre outros (Giffinger et al., 2007).

Apesar dos progressos na Mobilidade Urbana, problemas elementares ainda são prevalentes, tais como os danos causados ao meio ambiente em razão da emissão de poluentes bem como os impactos na qualidade de vida, tais como o tempo de comutação, a segurança e o conforto. Na evolução para uma Mobilidade Inteligente, a participação humana ativa é gradativamente substituída por sistemas e tecnologias inteligentes que colaboram para uma autonomia completa de transporte que, em última instância, tornam a função do motorista humano obsoleta (Kirwan and Zhiyong, 2020). Neste percurso evolutivo é natural, portanto, pensar primeiramente em soluções inteligentes que apoiem as pessoas para uma mobilidade mais eficiente e segura.

A detecção de semáforos, por exemplo, é uma tarefa importante para sistemas de assistência ao motorista e também para o desenvolvimento de veículos autônomos (Kim et al., 2018) objetivos importantes no contexto de Mobilidade Inteligente. Esta tarefa reside no âmbito da Visão Computacional (VC) e compreende a localização do semáforo e a classificação do seu estado, as quais são essenciais para uma posterior tomada de decisão. Há que se salientar que em contextos realistas há diversos desafios que não podem ser ignorados no desenvolvimento de soluções inteligentes para a detecção de semáforos, os quais compreendem variações no tamanho, posicionamento e luminosidade, condições de tempo e até mesmo ambiguidades com outros objetos urbanos (Behrendt et al., 2017).

Diversas soluções na literatura para o problema de detecção de semáforos utilizando diferentes estratégias para abordar a tarefa. Os primeiros trabalhos consideram técnicas clássicas de VC para extração de características, tais como forma e cor, seguidas da classificação com modelos de *Machine Learning*. Tais soluções mostram-se limitadas ao detectarem apenas semáforos de tamanhos e modelos específicos sob *backgrounds* pré-determinados, por exemplo (Diaz et al., 2015). Com o advento de *Deep Learning*, houve uma melhora no desempenho dessa tarefa sob condições menos controladas, considerando o paradigma do Aprendizado Supervisionado e arquiteturas de Redes Neurais Convolucionais Profundas (CNNs, do inglês *Convolutional Neural Networks*) e CNNs Regionais (R-CNNs, do inglês *Regional CNNs*) (Possatti et al., 2019).

Em problemas práticos de detecção de objetos em tempo real, as R-CNNs da família YOLO (acrônimo para *You Only Look Once*), em particular, tem se mostrado eficientes a partir de exemplos oriundos de diferentes bases de dados (Behrendt et al., 2017; Jensen et al., 2017). Uma das explicações para este bom desempenho deve-se ao fato de tal família de modelos utilizar a abordagem *single-shot* para detecção, em que as caixas delimitadoras (*bounding boxes*) dos objetos são previstas simultaneamente com a classificação associada (Redmon et al., 2016). Visando atestar o desempenho das R-CNNs YOLO na tarefa de detecção de semáforos e classificação, este trabalho apresenta uma análise comparativa de variantes da família YOLOv5, YOLOv7 e YOLOv8 utilizando a base de dados *Bosch Small Traffic Light Dataset* (Behrendt et al., 2017).

2. TRABALHOS RELACIONADOS

Considerando os objetivos desse trabalho, serão consideradas nessa discussão apenas trabalhos da literatura desenvolvidos e voltados para a mesma base de dados aqui considerada, o que propicia uma comparação equânime. Tal base, denominada *Bosch Small Traffic Lights Dataset* (BOSCH) (Behrendt et al., 2017), foi elencada para o escopo deste trabalho por ser pública, gratuita, diversa, realística e suas imagens possuírem boa resolução. Uma particularidade dessa base de dados é que a mesma possui mais exemplos de teste que de treinamento, o que visa propiciar um ambiente mais desafiador de preparação dos modelos e aferição de sua capacidade de generalização.

A primeira solução disponível na literatura sobre a base de dados em questão é de autoria dos seus próprios proponentes, os quais utilizaram um modelo YOLO como extrator

de características para uma CNN, cujas camadas de classificação e saída foram especificadas pelos próprios autores. Por meio de experimentos, os autores reportaram uma acurácia de 99% e 95,1% nas partições de treinamento e teste, respectivamente, mas afirmaram a necessidade de melhorias, pois a métrica IoU mostrou-se particularmente baixa, com valores aferidos entre 0,3 e 0,5 (Behrendt et al., 2017).

Dois trabalhos na literatura foram identificados utilizando, além da base de dados BOSCH, a base de dados de apoio LISA *Traffic Lights Dataset* (Philipsen et al., 2015). O primeiro trabalho utilizou três abordagens distintas para detecção e reconhecimento de semáforos, dentre as quais se sobressaiu uma R-CNN com mAP de 56,31% no conjunto de dados BOSCH e 76,37% no LISA (Ennahhal et al., 2019). O segundo trabalho consistiu na proposição de uma Rede Neural De-Convolucional com perda de regressão focal, composta por três sub-redes: uma codificadora, uma decodificadora e uma detectora. A rede decodificadora é responsável por gerar um mapa de características detalhado, porém com poucas informações contextuais, e outro pouco detalhado, mas com muitas informações contextuais, para logo em seguida ambos serem repassados à camada de codificação na qual serão combinados e então submetidos à camada de detecção. O modelo proposto conseguiu um mAP de 7,19% ~ 42,03% e de 19,86% ~ 49,16% nas bases de dados BOSCH e LISA, respectivamente (Lee and Kim, 2019). É interessante ressaltar, entretanto, que em face de mais exemplos relevantes para o problema, tem-se o favorecimento do aprendizado, proporcionando um aumento de desempenho geral.

A base de dados mencionada foi também utilizada na terceira etapa de uma competição ocorrida em 2021 intitulada *Data Science Challenge*, promovida pelo Instituto Tecnológico de Aeronáutica. A equipe LSI *Data Team* fez uso do modelo YOLOv5 e obteve um mAP de 0,56, as quais culminaram na classificação em segundo lugar nacional (Yonekura et al., 2021). Uma das principais motivações deste trabalho foi dar continuidade a este desenvolvimento e viabilizar experimentos para aferir se havia melhoria de desempenho perante novas versões da YOLO.

3. MATERIAL E MÉTODOS

O problema considerado no escopo deste trabalho foi abordado como uma tarefa de detecção mediante Aprendizado Supervisionado com R-CNNs da Família YOLO. A infraestrutura computacional utilizada consistiu em um servidor com Intel(R) Core(TM) i7-8700 CPU @ 3,20 GHz, 32 GB de memória principal, 2,4 TB de memória secundária e 2 GPUs NVIDIA GeForce GTX 1080 Ti com 11 GB de VRAM para aceleração em *hardware* do treinamento. Os dados experimentais, modelos e a avaliação de desempenho da tarefa considerada são descritos detalhadamente nas subseções a seguir.

3.1 Dados Experimentais

O conjunto de dados utilizados no experimento é proveniente do *Bosch Small Traffic Lights Dataset* (BOSCH) (Behrendt et al., 2017), que possui originalmente 13.425

imagens captadas em duas localizações diferentes, *Francisco Bay Area* e *Palo Alto*, na Califórnia. Neste *dataset* há cerca de 24.000 anotações que incluem as *bounding boxes* e os estados de cada semáforo. A base de dados é disponibilizada de forma particionada entre treino e teste, o quantitativo de cada partição e a sua respectiva distribuição está descrita na Tabela 1.

Tabela 1. Distribuição dos dados no BOSCH *dataset*.

	Quantidade de imagens	Percentual	Anotações	Percentual
Treino	5.093	38%	10.756	44%
Teste	8.334	62%	13.486	56%

Os dados pertencentes à partição de treino foram distribuídos de maneira aleatória e disjunta em duas partições de treino e validação, com 80% e 20% dos exemplos, respectivamente. A descrição deste particionamento pode ser visualizada na Tabela 2.

Tabela 2. Distribuição dos dados nas partições de treino e validação no *dataset* BOSCHv1.

	Quantidade de imagens	Anotações
Treino	3566	7611
Validação	1527	3144

A partição de validação é utilizada pela família YOLO como forma de ajustar os hiperparâmetros e avaliar o desempenho durante o treinamento

As classes presentes em cada partição giram em torno do padrão conhecido para semáforos urbanos: *Red* (Vermelho), *Yellow* (Amarelo) e *Green* (Verde), mas devido à frequência de captura da câmera aliada à taxa de atualização do semáforo, muitos sinais aparentam estar desligados, nesse caso optou-se pela criação da classe *off*. Além disso, em determinados exemplos, havia rótulos mais específicos que também incluíam direções, tais como *RedLeft* (Vermelho à Esquerda) e *GreenRight* (Verde à Direita).

Ao examinar o quantitativo de exemplos de semáforos com direção, percebeu-se que este era extremamente desbalanceado e escasso para algumas classes na partição de treinamento. Para prover experiência relevante ao modelo e propiciar a generalização na partição de testes, a especificação de direção foi eliminada, mantendo-se apenas o rótulo padrão do semáforo no tocante ao seu estado. Esse quantitativo nas partições de treino e teste é detalhado na Tabela 3.

Tabela 3. Distribuição de rótulos nas partições de treinamento e teste do BOSCH.

Classe	Treinamento		Teste	
	Quantidade	em %	Quantidade	em %
Green	5421	50,4	7569	56,12
Red	4164	38,7	5321	38,78
Yellow	444	4,1	154	1,14
off	726	6,8	442	3,27

Ao considerar um motorista humano, os semáforos que ocupam pouco espaço no campo visual podem também ser difíceis de classificar, necessitando de uma aproximação

para tomada de decisão quanto ao seu estado. No *dataset* original verificou-se muitos semáforos pequenos conforme critério do MS COCO (a partir de 5% da área total), conforme exibido na Tabela 4, o que levantou o questionamento se estes foram rotulados em tempo real ou *a posteriori*.

Tabela 4. Detalhamento dos semáforos quanto ao tamanho na partição de treinamento do BOSCH.

Classes	Pequeno	Médio	Grande	Total
Green	4734	681	6	5448
Red	3861	302	1	4164
Yellow	410	34	0	444
Off	659	67	0	726
Total	9664	1084	7	10782

Dada a suposição previamente mencionada e também a dificuldade dos modelos de detecção aprenderem padrões de objetos arbitrariamente pequenos distinguindo-os de ruído ou do *background*, avaliou-se a hipótese de treinar os modelos apenas com semáforos médios e grandes, o que diminui os custos computacionais, no intuito de aferir se tal estratégia de aprendizado é eficaz e eficiente para generalização posterior perante objetos pequenos. Como consequência, foi produzida uma versão modificada da partição de treino do BOSCH, a qual será BOSCHv2, apenas com objetos médios e grandes, com descrição estatística conforme Tabela 5 e distribuição qualitativa conforme a Tabela 6. Enfatiza-se que não houve alterações na partição de testes.

Tabela 5. Detalhamento dos semáforos quanto ao tamanho na partição de treinamento do BOSCHv2.

	Mínima	Média	Mediana	Máxima
Largura	16,76	29,39	26,84	98,0
Altura	23,75	63,10	58,23	207,00
Área	1.024,42	1.993,72	1.543,58	20.286,00

Tabela 6. Distribuição de rótulos nas partições de validação e treinamento do BOSCHv2.

Classe	Validação		Treinamento	
	Quantidade	em %	Quantidade	em %
Green	220	64,52	687	63,0
Red	87	25,51	303	27,8
off	25	7,33	67	6,1
Yellow	9	2,64	34	3,1

3.2 Modelos, Parâmetros e Hiperparâmetros

No escopo deste trabalho serão consideradas diferentes versões da arquitetura YOLO, a citar: versões 5(Jocher et al., 2020), 7(Wang et al., 2022) e 8(Jocher et al., 2022), cada uma com um número diferente de parâmetros. Em todas as versões utilizadas, foi aplicada a técnica de *Early Stopping* que verifica se houve melhoria na generalização no conjunto de validação; persistência em disco do melhor conjunto de pesos obtidos durante o treinamento perante

o conjunto de validação. Em relação aos hiperparâmetros, os valores padrões foram utilizados exceto em relação ao tamanho do *batch* e o número de épocas que variaram de acordo com cada modelo.

Para a YOLOv5 serão avaliadas as versões *Nano*, *Small* e *Medium* com os seguintes parâmetros, hiperparâmetros e características:

Tabela 7. Descrição dos parâmetros das arquiteturas YOLOv5's utilizadas.

Modelo	Parâmetros	Camadas	Batch	Épocas
Yolov5 Nano	3.107.836	281	16	500
Yolov5 Small	7.041.205	214	16	500
Yolov5 Medium	20.899.605	291	16	500

Para a YOLOv7 serão avaliadas as versões *YOLOv7*, *YOLOv7-X* e *YOLOv7-W6* com os seguintes parâmetros, hiperparâmetros e características:

Tabela 8. Descrição dos parâmetros das arquiteturas YOLOv7's utilizadas.

Modelo	Parâmetros	Camadas	Batch	Épocas
Yolov7	36.900.000	105	24	600
Yolov7-X	71.300.000	121	16	600
Yolov7-W6	70.400.000	122	16	600

Devido a complicações causadas pela limitação do *hardware* utilizado, tornou-se necessário a utilização de *batches* diferentes dos que foram utilizados anteriormente, com o intuito de incrementar a velocidade de treinamento.

Para a YOLOv8 serão avaliadas as versões *Nano*, *Small* e *Medium* com os seguintes parâmetros, hiperparâmetros e características:

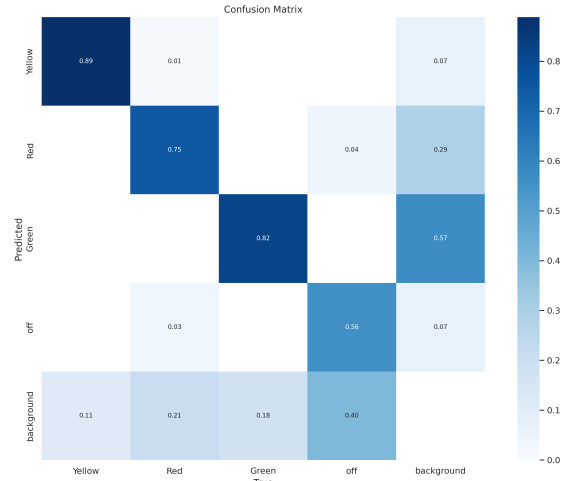
Tabela 9. Descrição dos parâmetros das arquiteturas YOLOv8 utilizadas.

Modelo	Parâmetros	Camadas	Batch	Épocas
Yolov8 Nano	3.011.043	225	16	600
Yolov8 Small	11.135.987	225	16	600
Yolov8 Medium	25.856.899	295	16	600

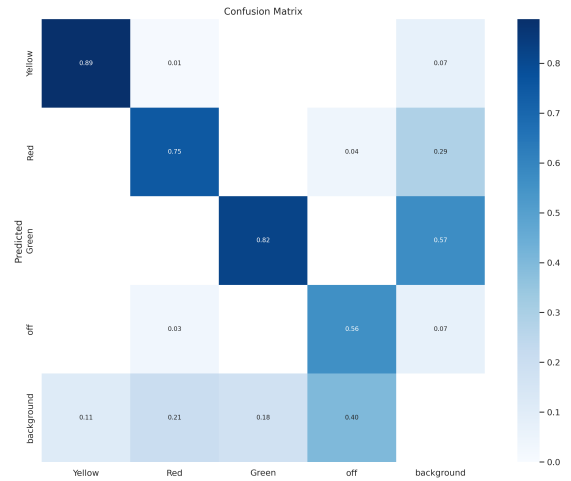
4. RESULTADOS E DISCUSSÃO

Uma vez estabelecida a metodologia experimental, partiu-se para a execução do treinamento e teste dos modelos, este último perante a partição original da base de dados BOSCH. Os resultados obtidos pelos diferentes modelos e versões de arquiteturas da Família YOLO encontram-se detalhados na Tabela 10.

Para analisar as métricas de desempenho obtidas, os modelos da família YOLO foram validados em suas respectivas partições de teste tanto para o *dataset* BOSCHv1 quanto *dataset* BOSCHv2, concluiu-se que as variantes da arquitetura YOLOv8, especificamente as versões *Nano* e *Small*, obtiveram um bom desempenho, com um mAP@0.5 de 0,826 e 0,851, respectivamente. Nas matrizes de confusão ilustradas nas Figuras 1a e 1b percebe-se a prevalência de dados na diagonal principal, atestando uma boa eficiência dos modelos referidos no tocante à tarefa considerada.



(a) Nano



(b) Small

Figura 1. Matrizes de confusão de modelos YOLOv8.

Apesar do *dataset* de treinamento BOSCHv2 possuir uma quantidade reduzida de exemplos, todos os modelos treinados sobre o mesmo demonstraram bom desempenho na etapa posterior de testes. Este achado demonstra que a diminuição do ruído, por meio da redução das *bounding boxes* pequenas, foi benéfica para a melhora não apenas das métricas obtidas, como também para a redução do tempo de treinamento, fazendo com que os modelos YOLOv8 *Small* e *Medium* atingissem o *Early Stopping* nas épocas 222 e 270, com um valor padrão de paciência igual a 50, e tempo de treinamento entre 2,5 e 3 horas, respectivamente. Os resultados obtidos neste cenário em específico comprovam que a Família YOLO no geral, possui boas técnicas de generalização e *data augmentation* possibilitando um boa diversidade de dados para o treinamento. Alguns exemplos ilustrativos de detecção são mostrados nas Figuras 2a e 2b.

É importante notar que os modelos mais antigos mostraram-se mais flexíveis em relação aos dados de treinamento, como aqueles da YOLOv5, os quais demonstraram um desempenho superior notável quando comparado aos outros modelos no mesmo *dataset* BOSCHv1, evidenciando que melhorias implementadas com foco em *datasets* com ima-

Tabela 10. Síntese dos resultados experimentais.

Dataset	Modelo	Tempo de Treinamento	Precisão	Revocação	F-Score	mAP@0.5
BoschV1	YOLOv5 Nano	4h:40min	0.717	0.562	0.630	0.613
	YOLOv5 Small	4h:14min	0.735	0.596	0.658	0.627
	YOLOv5 Medium	3h:44min	0.719	0.611	0.660	0.642
	YOLOv7	10h:55min	0.454	0.248	0.321	0.255
	YOLOv7-X	19h:44min	0.656	0.467	0.546	0.481
	YOLOv7-W6	18h:12min	0.472	0.385	0.424	0.374
	YOLOv8 Nano	4h:24min	0.702	0.476	0.567	0.523
	YOLOv8 Small	5h:11min	0.682	0.480	0.563	0.549
	YOLOv8 Medium	6h:35min	0.705	0.485	0.575	0.556
BoschV2	YOLOv5 Nano	2h:50min	0.697	0.599	0.644	0.635
	YOLOv5 Small	4h:9min	0.763	0.566	0.649	0.644
	YOLOv5 Medium	4h:9min	0.761	0.581	0.658	0.653
	YOLOv7	10h:10min	0.866	0.527	0.655	0.649
	YOLOv7-X	17h:58min	0.740	0.562	0.639	0.648
	YOLOv7-W6	15h:15min	0.800	0.606	0.689	0.657
	YOLOv8 Nano	2h:20min	0.707	0.824	0.761	0.826
	YOLOv8 Small	2h:55min	0.769	0.779	0.774	0.851
	YOLOv8 Medium	4h:34min	0.860	0.708	0.777	0.821



(a) YOLOv8 Nano



(b) Small

Figura 2. Exemplos de detecção com modelos YOLOv8.

gens com características variadas diferentes das presentes no *dataset COCO* (Lin et al., 2015) também incrementaram as métricas obtidas.

5. CONSIDERAÇÕES FINAIS

O objetivo desta proposta de trabalho consiste em avaliar o uso de CNNs da família YOLO para a tarefa de detecção de semáforos e classificação a partir da base de dados *Bosch Small Traffic Light Dataset* (Behrendt et al., 2017) que

possui mais de 13.000 imagens captadas. Durante a primeira etapa, foi utilizado a arquitetura YOLOv5 (Jocher et al., 2020), especificamente suas variantes *Nano*, *Small* e *Medium* e considerou-se duas bases de dados, a *Bosch Small Traffic Light* e uma derivação, denominada BOSCHv2. Os melhores resultados foram obtidos pelas arquiteturas YOLOv5 *Small* e YOLOv5 *Medium*, com um mAP@0.5 de 0,627 e 0,642, respectivamente.

Dando sequência aos experimentos realizados inicialmente, outras versões da família YOLOv7 e YOLOv8 foram implementadas e avaliadas e verificou-se que o desempenho dos modelos *YOLOv8 Nano* e *YOLOv8 Small* foi superior aos demais, com um mAP@0.5 de 0,826 e 0,851, respectivamente, mesmo com uma quantidade de exemplos limitada no *dataset BOSCHv2*, evidenciando a capacidade de generalização da família YOLOv8. Desta forma, apresenta-se como uma solução competitiva para o problema de detecção semafórica.

Em trabalhos futuros, pretende-se ampliar o número de modelos avaliados da família YOLO e realizar novas experimentações. Além disso, considera-se a captação de novos exemplos para a ampliação do *dataset* consolidando um conjunto de dados mais robusto e abrangente e rebalancear a quantidade de imagens obtidas de forma mais eficiente.

AGRADECIMENTOS

Os autores agradecem o apoio material do Laboratório de Sistemas Inteligentes (LSI) da Universidade do Estado do Amazonas (UEA).

REFERÊNCIAS

- Behrendt, K., Novak, L., and Botros, R. (2017). A deep learning approach to traffic lights: Detection, tracking, and classification. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Singapore, Singapore.
- Diaz, M., Cerri, P., Pirlo, G., Ferrer, M.A., and Impedovo, D. (2015). A survey on traffic light detection. In *New Trends in Image Analysis and Processing - ICIAP 2015*

- Workshops*, Lecture notes in Computer Science, 201–208. Springer International Publishing, Itália.
- Ennahhal, Z., Berrada, I., and Fardousse, K. (2019). Real time traffic light detection and classification using deep learning. In *2019 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 1–7. doi:10.1109/WINCOM47513.2019.8942446.
- Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Milanović, N., and Meijers, E. (2007). *Smart cities - Ranking of European medium-sized cities*.
- Jensen, M.B., Nasrollahi, K., and Moeslund, T.B. (2017). Evaluating state-of-the-art object detector on challenging traffic light data. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE.
- Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu, Z., Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, Unglv-KitDe, Sonck, V., tkianai, yxNONG, Skalski, P., Hogan, A., Nair, D., Strobel, M., and Jain, M. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. doi:10.5281/zenodo.7347926. URL <https://doi.org/10.5281/zenodo.7347926>.
- Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, Hogan, A., lorenzomamma, yxNONG, AlexWang1900, Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Ingham, F., Frederik, Guilhen, Hatovix, Poznanski, J., Fang, J., Yu, L., changyu98, Wang, M., Gupta, N., Akhtar, O., PetrDvoracek, and Rai, P. (2020). ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements. Disponível em <https://doi.org/10.5281/zenodo.4154370>. Acesso em 27 de agosto de 2023.
- Kim, H.K., Park, J.H., and Jung, H.Y. (2018). An efficient color space for deep-learning based traffic light recognition. *Journal of advanced transportation*, 2018.
- Kirwan, C. and Zhiyong, F. (2020). *Smart cities and artificial intelligence*. Smart Cities. Elsevier Science Publishing, Philadelphia, PA.
- Kon, F. and Santana, E.F.Z. (2016). *Cidades Inteligentes: Conceitos, plataformas e desafios*, volume 17. SBC, Porto Alegre. Disponível em <https://doi.org/10.5753/sbc.6.1>.
- Lee, E. and Kim, D. (2019). Accurate traffic light detection using deep neural network with focal regression loss. *Image and Vision Computing*, 87, 24–36. doi:<https://doi.org/10.1016/j.imavis.2019.04.003>. URL <https://www.sciencedirect.com/science/article/pii/S0262885619300538>.
- Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., and Dollár, P. (2015). Microsoft coco: Common objects in context.
- Philipsen, M.P., Jensen, M.B., Møgelmoose, A., Moeslund, T.B., and Trivedi, M.M. (2015). Traffic light detection: A learning algorithm and evaluations on challenging dataset. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2341–2345. doi:10.1109/ITSC.2015.378.
- Possatti, L.C., Guidolini, R., Cardoso, V.B., Berriel, R.F., Paixao, T.M., Badue, C., De Souza, A.F., and Oliveira-Santos, T. (2019). Traffic light recognition using deep learning and prior maps for autonomous cars. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Budapest, Hungary.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, Estados Unidos.
- Song, H., Srinivasan, R., Sookoor, T., and Jeschke, S. (2017). *Smart cities*. Standards Information Network.
- Wang, C.Y., Bochkovskiy, A., and Liao, H.Y.M. (2022). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Disponível em <https://doi.org/10.48550/arxiv.2207.02696>. Acesso em 27 de agosto de 2023.
- Yonekura, D.C., de Oliveira, M.V., and Mendes, L.M. (2021). Lsi data team no data science challenge at ita edição 2021. Disponível em <https://www.youtube.com/watch?v=RmodnKr65aI>. Acesso em 27 de agosto de 2023.