MOTORIZADO PARA O CAMPUS UNIVERSITÁRIO DA UNICAMP

Palavras-Chave: TRANSPORTE MOTORIZADO, CAMPUS SUSTENTÁVEL, YOLO

Autores(as):

GABRIEL BRUNO CASAGRANDE, FECFAU - UNICAMP

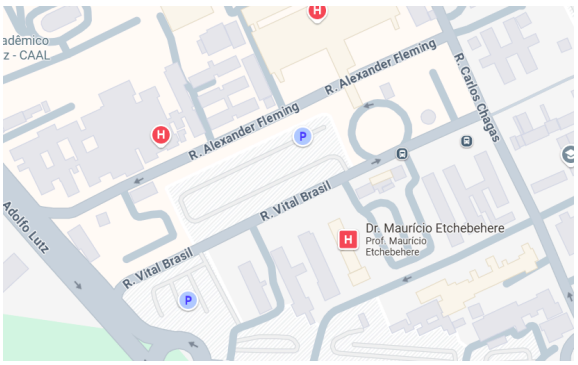
Prof(ª). Dr(ª). JANAÍNA ANTONINO PINTO (Orientadora), FECFAU - UNICAMP

Ma. ESTER DIVIESO ROMAN RODRIGUES (Coorientadora), FECFAU - UNICAMP

INTRODUÇÃO:

No Brasil, o setor de transportes responde pela segunda maior parcela de emissões de poluentes — e é o principal emissor dentro do setor de energia, segundo o Inventário Nacional SEEG (2021). Campi universitários, como a Unicamp, funcionam como microcidades: concentram milhares de viagens diárias, mas ainda carecem de métodos padronizados para quantificar padrões de deslocamento e emissões. Entender esses fluxos é condição essencial para que iniciativas de Campus Sustentável sejam efetivas.

Para medir esses fluxos com precisão, foi adotado a família de modelos de Visão Computacional YOLO (You Only Look Once), capaz de detectar objetos em tempo real com alta acurácia e baixo custo computacional. O YOLO divide a imagem em uma grade, atribui probabilidades a cada célula e devolve, em uma única etapa, a localização e a classe dos objetos (Redmon et al., 2016). O uso do YOLO nesta pesquisa possui, como objetivo principal, a incumbência de contar quantos veículos trafegam diariamente pela área da saúde da unicamp, especialmente o quarteirão entre as ruas Alexander Fleming, Carlos Chagas, Vital Brasil e Avenida Adolfo Lutz que foram o objeto de estudo da iniciação científica.



Além da detecção quadro a quadro, aplicou-se rastreamento com filtro de Kalman para evitar contagens duplicadas, aumentando a confiabilidade dos resultados observados em testes de campo no entorno do Hospital de Clínicas da Unicamp. Os dados processados podem ser

temáticas de emissão, o que viabiliza análises espaço-temporais robustas.

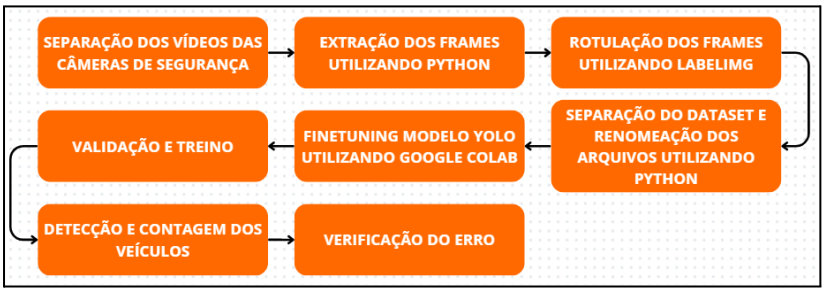
Em síntese, a combinação YOLO + rastreamento permite construir uma rede de dados de transporte que identifica horários de pico, perfis de frota e pontos críticos de emissão — fornecendo subsídios diretos para políticas de mobilidade e intervenções viárias mais sustentáveis no campus.

METODOLOGIA:

Para realizar a contagem e classificação dos veículos com precisão, fez-se necessário um retreinamento do modelo YOLO v11. Para isso, foram necessárias uma série de etapas que envolveram desde a preparação do dataset até o ajuste final do modelo. A seguir, são descritas as principais fases do processo metodológico:

1. Extração de frames dos vídeos (*extratordeframes.py*)

Inicialmente, foi realizada a extração de frames a partir dos vídeos concedidos das câmeras de segurança do campus obedecendo todas as normas legais. Para isso, desenvolveu-se um script em Python denominado *extratordeframes.py*, que permitiu capturar frames em intervalos de tempo definidos. Em geral, foi adotada a extração de um frame a cada 30 quadros, podendo esse intervalo ser ajustado conforme a necessidade de maior amostragem de um mesmo veículo. Os arquivos extraídos foram nomeados sequencialmente a partir de 0.png, garantindo uma organização sistemática do dataset.



2. Rotulagem das imagens com *LabelImg*

As imagens extraídas foram rotuladas manualmente por meio do software open source *LabelImg*. Esse processo é essencial para o treinamento supervisionado do modelo YOLO, pois fornece as coordenadas dos objetos de interesse (veículos, pessoas etc.) em cada imagem. O *LabelImg* gera, para cada imagem rotulada, um arquivo .txt com a classe do objeto e suas coordenadas normalizadas. Dessa forma, cada imagem

mesmo identificador numérico no nome do arquivo.

3. Padronização das classes rotuladas *(alteradordetxt.py)*

Considerando que o trabalho de rotulagem foi realizado em equipe, houve divergências na ordenação numérica das classes nos arquivos .txt gerados. Para resolver essa inconsistência e garantir a uniformidade do dataset, foi desenvolvido o script *alteradordetxt.py*, responsável por substituir os índices das classes nos arquivos de texto com base em uma lista padronizada: [car, person, van, bus, motorcycle, bicycle] → [0, 1, 2, 3, 4, 5].

4. Renomeação dos arquivos (*renomeador.py*)

Durante a construção coletiva do dataset, diferentes pastas foram criadas com arquivos nomeados de forma semelhante (iniciando em 0.png, 1.png, etc.). Para possibilitar a união de todos os conjuntos de dados em um único diretório, foi necessário renomear os arquivos a fim de evitar sobreposição de nomes. O script *renomeador.py* foi desenvolvido com essa finalidade, somando um valor offset ao nome de cada arquivo com base no maior índice da pasta anterior.

5. Separação em subconjuntos de treino e validação (*split-code.py*)

Com o dataset unificado e padronizado, realizou-se a divisão entre dados de treino e validação. O modelo YOLO exige uma estrutura específica de diretórios, com pastas separadas para imagens e arquivos de rotulagem, cada uma subdividida em *train* e *val*. Para automatizar essa tarefa, foi criado o script *split-code.py*, que distribuiu os dados aleatoriamente em proporção média de 75% para treino e 25% para validação.

6. Treinamento do modelo no Google Colab

A etapa final consistiu no treinamento do modelo YOLO v11, utilizando a infraestrutura do Google Colab. O dataset foi carregado no Google Drive e complementado com um arquivo *data.yml*, que contém a definição das classes a serem aprendidas pelo modelo. Os experimentos foram realizados com variações no número de épocas (5, 10 e 20), permitindo avaliar o desempenho do modelo em diferentes estágios de aprendizado. Diversos datasets foram testados, com alterações na quantidade de imagens por classe, buscando identificar a melhor configuração para a tarefa de detecção de veículos.

RESULTADOS E DISCUSSÃO:

Foi desenvolvida uma pipeline computacional completa para análise de tráfego no campus da Unicamp, implementada em Python e articulada em módulos que automatizam

A solução contempla, de forma integrada, (i) extração de frames a partir de vídeos de câmeras do campus, com taxa de amostragem ajustável conforme a densidade de eventos e as necessidades de anotação; (ii) separação e renomeação sistemática de arquivos para organizar imagens e rótulos em formato compatível com YOLO, mitigando conflitos de nomenclatura em bases extensas e padronizando as divisões de treino/validação/teste; (iii) re-treinamento do modelo YOLO em ambiente Google Colab; e (iv) módulo de inferência e contagem que combina detecção com rastreamento por filtro de Kalman, evitando duplicações e consolidando trajetórias para produzir contagens por intervalo de tempo e por tipo de veículo.

A pipeline foi validada em ensaio piloto com vídeos reais do entorno do Hospital de Clínicas da Unicamp, confirmando a viabilidade técnica do método no contexto do campus. Observou-se a detecção consistente de classes como *car, person, bicycle, van, bus and motorcycle* e a exibição da confiança do modelo durante o processamento, evidenciando a robustez da etapa de detecção. O módulo de rastreamento permitiu reduzir contagens redundantes, garantindo a fidedignidade das detecções geradas.

O retreinamento realizado da rede neural YOLO foi essencial para que classes de objetos que não são nativos do algoritmo de detecção, como por exemplo as vans, fossem detectadas corretamente. Logo, para ter uma contagem precisa das vans que trafegam diariamente pelo campus da UNICAMP, fez-se necessário um dataset robusto com imagens em diferentes ângulos de visão do tipo de automóvel mencionado.

Ao consolidar detecção, rastreamento e integração espacial, a pipeline viabiliza a construção de uma rede de dados de transporte motorizado no âmbito do Campus Sustentável da Unicamp, oferecendo insumos diretamente aplicáveis ao planejamento da mobilidade interna.

CONCLUSÕES:

Este trabalho demonstrou a viabilidade e a utilidade de uma pipeline computacional integrada para observação e análise do tráfego motorizado no campus da Unicamp, articulando detecção automática com YOLO e rastreamento por filtro de Kalman. Do ponto de vista técnico, avançou-se desde a preparação sistemática do dataset até a contagem final por classe veicular, entregando uma infraestrutura reprodutível e escalável que reduz esforço manual, padroniza processos e melhora a confiabilidade das detecções geradas.

Os resultados obtidos sustentam a construção de uma rede de dados de mobilidade no campus e abrem caminho para análises espaciais e ambientais de maior resolução. Como

no ângulo de visão das câmeras de segurança, por isso, para obter a precisão máxima no retreinamento, faz-se necessário que as imagens do dataset utilizado sejam exclusivamente da câmera de segurança que posteriormente será aplicada a contagem de veículos. Em caso de necessidade de contagem de veículos em mais de uma câmera, é interessante que sejam realizados um retreinamento para cada câmera desejada, obtendo assim modelos diferentes que serão mais precisos ao serem aplicados em ângulos específicos.

Em síntese, a pesquisa entrega um arcabouço técnico e metodológico que conecta visão computacional e rastreamento para informar decisões de planejamento e gestão da mobilidade no campus. A abordagem é replicável, pode ser expandida a outros setores da universidade e fornece um ponto de partida sólido para estudos longitudinais de emissões, avaliação de políticas de redução de tráfego motorizado e desenho de intervenções que promovam um ambiente universitário mais seguro, eficiente e sustentável.

BIBLIOGRAFIA:

CETESB - Companhia Ambiental do Estado de São Paulo. **Emissões veiculares no estado de São Paulo 2020.** São Paulo: CETESB, 2022. Disponível em: https://cetesb.sp.gov.br/veicular/relatorios-epublicacoes/ ISBN 978-65-5577-031-5.

MACHADO, Lindinalva Candido. **A gestão do projeto campus sustentável na Unicamp.** Sínteses: Revista Eletrônica do SimTec, Campinas, SP, n. 7, p. e019109, 2019. Disponível em: https://econtents.bc.unicamp.br/inpec/index.php/simtec/article/view/11566.

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, p. 779–788, 2016.

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, p. 779–788, 2016.

SEEG - Sistema de Estimativas de Emissões e Remoções de Gases de Efeito Estufa. **Análise das emissões brasileiras de e suas implicações para as metas climáticas do Brasil 1970–2020,** 2021.

YIN, Robert K. **Estudo de Caso: Planejamento e Métodos.** Bookman Editora, 2015.