

# Detecção de objetos: relatório do trabalho prático

Felipe Bona, João Martinho

19 de maio de 2025

## 1 Montagem e preparação da base

A base de dados é um clipe de 22 segundos de uma rua movimentada na Coreia do Sul colhida em um banco de vídeos sem direitos autorais. O vídeo, depois de baixado, foi convertido para o formato `.mp4`, e o modelo utilizado foi o YOLOv8 pré-treinado, para incluir suporte a pedestres e bicicletas. O ambiente Python foi configurado com as bibliotecas `ultralytics`, `opencv-python`, `numpy` e `pandas`. Além de pedestres e bicicletas, o modelo inclui carros, caminhões, ônibus e vans.

## 2 Modelo e arquitetura

Foi escolhido o modelo nano (YOLOv8n), pela sua eficiência e precisão em tarefas de detecção em tempo real. Utilizamos pesos pré-treinados do `yolov8n.pt`. Configuramos as classes “car”, “truck”, “bus” e “van”, e utilizamos a generalização do modelo COCO para a detecção de pedestres e bicicletas.

## 3 Treinamento

Não fizemos treinamento adicional, preferindo confiar na robustez do modelo COCO. No entanto, se fosse necessário, o treinamento seria feito com:

- Divisão do dataset em treino/validação/teste (80/10/10);
- Ajuste fino dos pesos com os vídeos do UA-DETRAC;
- Hiperparâmetros: Learning rate = 0.001, batch size = 16, épocas = 50.

## 4 Classificação e testes

O código-fonte (`trabalho.py`) realiza a leitura do vídeo processando quadros através do OpenCV e aplica o YOLOv8 para detectar objetos em cada quadro, usando IDs para evitar a contagem dupla e incrementando contadores por classe. A saída gerada é um arquivo `.csv` contendo os resultados para cada quadro do vídeo.

## 4.1 Testes

A implementação foi testada usando o clipe de tráfego urbano já mencionado. O arquivo de saída contém a contagem de veículos para cada quadro do vídeo, separados em classes distintas.

## 5 Discussão dos resultados

O modelo YOLOv8n é adequado para vídeos em tempo real e mostra alta acurácia na detecção de veículos, mas ainda há imperfeições, como a confusão de certos tipos de veículos (ônibus e caminhões, por exemplo) em alguns casos. Possíveis melhorias futuras incluem o ajuste com anotações adicionais para pedestres e bicicletas e a otimização com o uso do YOLOv8s, bem como a exploração de outras amostras de dados.

## 6 Conclusão

O trabalho demonstrou a aplicação do YOLOv8 para detecção e contagem de veículos em vídeos de tráfego urbano. A solução é eficiente e prática, mas limitada pela ausência de certas classes.