UNIVERSIDADE REGIONAL DE BLUMENAU

PROCESSAMENTO DE IMAGEM

PROFESSOR AURÉLIO HOPPE

ACADÊMICOS: Alani Rigotti de Oliveira, Ari Júnior, Felipe Augusto de Carvalho de Araújo e Luigi Marchetti

## TRABALHO 3 – DETECÇÃO DE OBJETOS

Processamento de imagem é uma área da computação dedicada à extração, interpretação e manipulação de informações presentes em imagens digitais. Essa tecnologia tem sido amplamente aplicada em diversos setores, como no desenvolvimento de sistemas inteligentes de trânsito, na seleção de grãos na agricultura de precisão, entre outros. Dentre essas aplicações, destaca-se o uso em ambientes urbanos, especificamente na análise de semáforos, cruzamentos e faixas de pedestres, elementos essenciais para o monitoramento e controle do tráfego.

Neste contexto, este trabalho tem como objetivo analisar vídeos de mobilidade urbana com ênfase na detecção de semáforos. Para isso, é desenvolvido com a linguagem de programação Python, utilizando a biblioteca **YOLO (You Only Look Once)** em específico as versões 8, 11n e 11m. Essa biblioteca utiliza uma rede neural convolucional (CNN) e, por meio de caixas delimitadoras e probabilidades de classe diretamente a partir de frames do vídeo ou imagem, realiza a detecção de objetos em tempo real. A escolha do Yolo foi motivada pela vantagem da alta precisão e desempenho na identificação de elementos visuais relevantes em ambientes complexos.

Figura 1: Metodologia utilizada para o desenvolvimento do presente trabalho.


(Fonte: elaborado pelos autores)

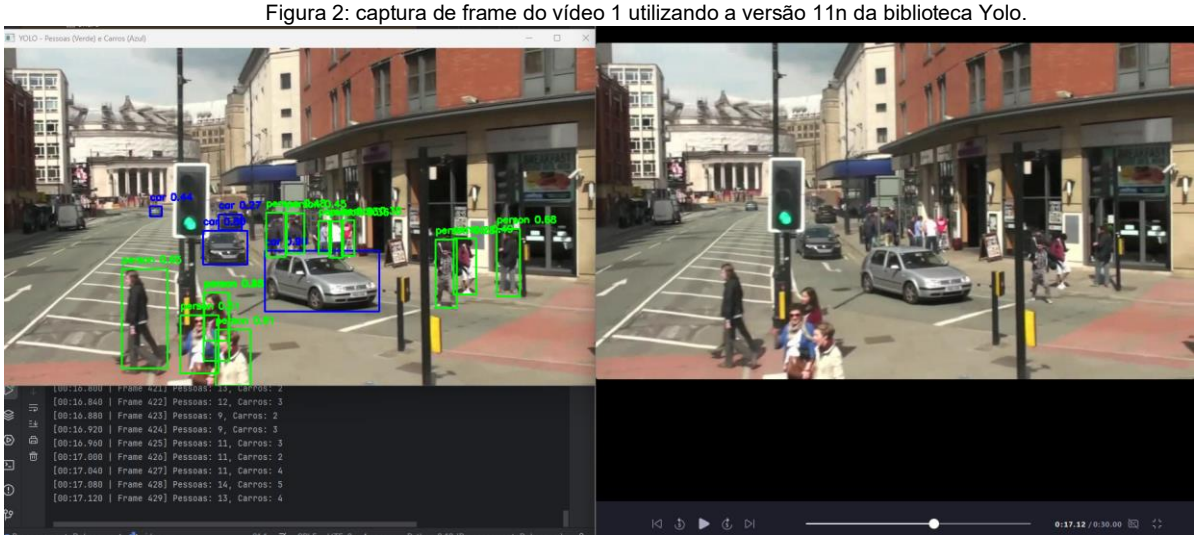
Para realizar o trabalho, o professor Aurélio Hoppe orientou a usar um modelo já treinado e ter como base a documentação do Yolo para o desenvolvimento do presente trabalho. Então, após isto, foi feito a configuração do ambiente de desenvolvimento PyTorch, biblioteca Yolo e, principalmente, o vídeo para aplicar o algoritmo desta biblioteca. Com isso, a partir do MOTChallenge, site que disponibiliza conjuntos de imagens e vídeos, foi escolhido dois vídeos. Um dos vídeos é mais simples, apenas com pessoas e carros para entendermos a biblioteca Yolo. E o outro mais específico, com fluxo de pessoas no período noturno.

Vídeo 1: https://motchallenge.net/vis/MOT17-13

Vídeo 2: https://motchallenge.net/vis/MOT17-04-SDP

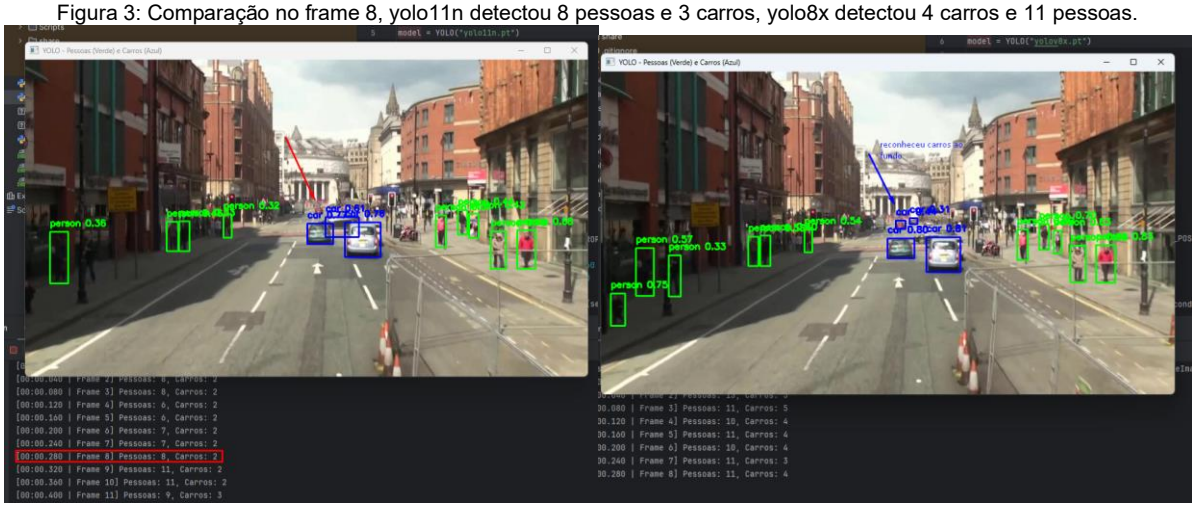
Os testes iniciaram com a aplicação da versão 11n da biblioteca Yolo no vídeo 1. Conforme na figura 2, podemos perceber que o resultado consistiu na identificação precisa de pessoas (person), adultos e crianças, e tipos de veículos, como caminhão (truck) e carro (car). Enquanto na versão 8x, além de identificar pessoas e veículos, identificou objetos mais distantes da câmera, conforme figura 3.

Figura 2: captura de frame do vídeo 1 utilizando a versão 11n da biblioteca Yolo.




(Fonte: elaborado pelos autores)

Figura 3: Comparação no frame 8, yolo11n detectou 8 pessoas e 3 carros, yolo8x detectou 4 carros e 11 pessoas.




(Fonte: elaborado pelos autores)

Para comparar melhor o desempenho da biblioteca Yolo em diferentes contextos de iluminação e objetos, foi utilizado o vídeo 2. Nesse segundo teste, foram utilizadas as versões 11n, 11x, 11m e 8x. Na comparação das versões, é possível observar que a detecção possui obstáculos como a baixa iluminação e sobreposição de objetos. Na versão 11x performance baixa (ainda melhor em relação a 8x), e a maior precisão de detecção de objetos, poucos falsos positivos, alto grau de confiança, porém ainda com certa dificuldade de detectar pessoas em aglomerações. Já o modelo 11m apresentou uma performance média, com pouca melhora na detecção em relação ao 11n.

Figura 4: captura de frame do vídeo 1 utilizando a versão 11x da biblioteca Yolo.


(Fonte: elaborado pelos autores)

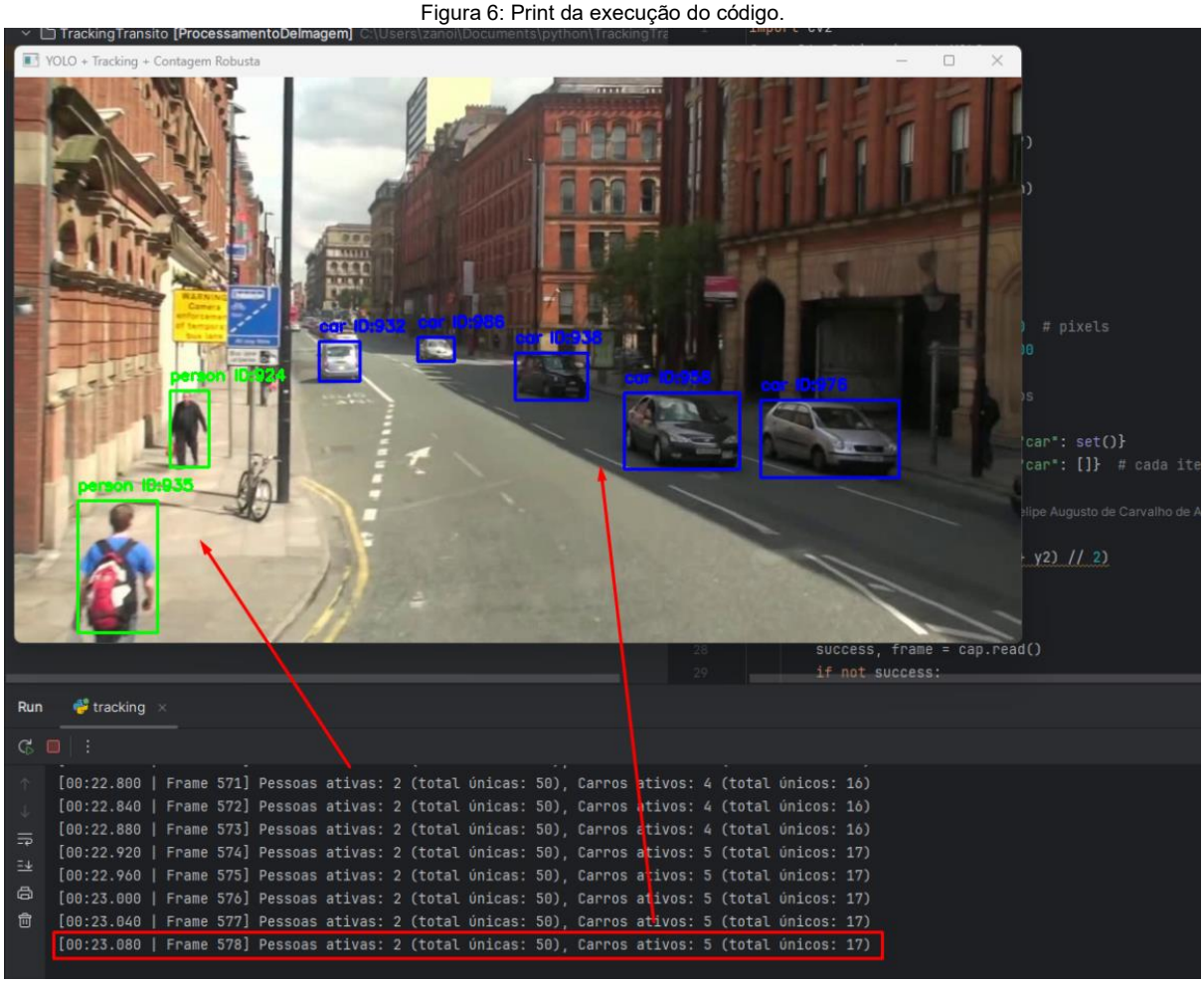
Figura 5: Captura de falso positivo utilizando a versão 11m.


(Fonte: elaborado pelos autores)

## Tracking e Contagem total

Implementamos a contagem total de pessoas e carros únicos em cada vídeo, utilizando o tracking do YOLO, que atribui um ID exclusivo a cada objeto detectado.

Figura 6: Print da execução do código.




(Fonte: elaborado pelos autores)

Durante os testes iniciais, observou-se que a contagem estava sendo superestimada devido à reincidência de detecções do mesmo objeto após curtos períodos de ausência, resultando em duplicações. Exemplo abaixo de recontagem de uma pessoa no vídeo 2, ocorreu após ela se posicionar atrás do poste e voltar.

Figura 7: Pessoa identificada pelo algoritmo


(Fonte: elaborado pelos autores)

Figura 8: Mesma pessoa da figura 6, mas identificada como outra pessoa


(Fonte: elaborado pelos autores)

Tentando minimizar isso todos os modelos mencionados (YOLO11n.pt, YOLO11x.pt, YOLO11m.pt, YOLOv8x.pt) utilizam Non-Maximum Suppression (NMS) como parte do pipeline de pós-processamento por padrão. São eliminadas bounding boxes duplicadas de um mesmo objeto no mesmo frame, selecionando a box com maior pontuação de confiança e descartando outras com alta sobreposição (IoU acima de um limiar, ex.: 0.5). Isso ajuda a garantir uma única detecção por objeto, reduzindo IDs duplicados. Não usa keypoints diretamente, mas suporta pipelines de rastreamento e pose estimation.

Para mitigarmos esse problema, foi implementada uma lógica de persistência temporal e espacial. Um objeto passou a ser considerado válido para contagem apenas se permanecesse visível por no mínimo 3 segundos, e não fosse detectado novamente na mesma região em um intervalo curto de tempo. Essa abordagem visou reduzir contagens duplicadas decorrentes de pequenas interrupções na detecção.

Após a validação da solução, o tempo mínimo foi ajustado para 1 segundo, com o objetivo de aumentar a sensibilidade do sistema frente a passagens rápidas, mantendo a precisão na exclusão de duplicatas. Dessa forma, o algoritmo chegou no número de 55 pessoas e 20 carros. Apesar de ainda haver duplicação, é importante ressaltar que no vídeo 1 a gravação utilizada é em movimento. Uma forma posterior de resolver isto seria uma gravação estática, ou seja, que a câmera esteja fixa e em um ângulo melhor. Além, também, da possibilidade de utilizar mais algoritmos para guardar o contexto de cada objeto, mas demandaria um custo operacional maior.

**Conclusão:**

Após os testes, concluímos que o modelo YOLO11n é a melhor escolha geral, pois oferece excelente desempenho com qualidade de detecção próxima ao YOLO11x. Para o caso da doutoranda Ana Lúcia, que busca detectar, classificar e contar pedestres, carros, bicicletas etc, em ruas, possivelmente usando múltiplas câmeras, a escolha do YOLO11n é vantajosa devido à sua eficiência e economia de recursos.

Caso se preze principalmente pela precisão seria recomendado o 11x, mas seria necessária uma máquina muito potente. Para referência, em nossos testes utilizamos uma máquina com um processador Ryzen 5 5600 com clock de 3.5 GHz e placa gráfica RTX 3060 com 3584 CUDA Cores tivemos uma média de 2 FPS (*frames per second*) em média com o modelo 11x e 20 FPS com o 11n.

Conseguimos obter uma contagem aproximada do valor real de carros e pessoas contadas por humano, com uma margem de erro para cima, pois pode reconhecer em alguns cenários específicos um objeto mais de uma vez, para evitar este cenário seria necessário a implementação de um algoritmo que guarde informações do objeto detectado, mas isto também aumentaria o custo de processamento.