Laboratório de Visão Computacional – EP02

Nome: Lucas Martinuzzo Batista

O segundo EP pede se conte quantos automóveis aparecem em um vídeo de estradas e avenidas e oferece 6 vídeos para o exercício.

A abordagem apresentada pelo professor em sala de aula “desenha” uma região de interesse onde as detecções são realizadas e, dentro desta região de interesse, cria uma linha imaginária que os veículos devem passar para serem contabilizados. Minha abordagem foi um pouco diferente. Eu criei um rastreador que acompanha a Bounding Box (BB) desde sua primeira aparição, e contabiliza as bounding boxes que apareceram ao menos 15 frames. Uma BB é considerada a mesma do quadro anterior, se no quadro anterior tiver aparecido uma bounding box com 50 pixels ou menos de distância (euclidiana) da atual.

Para isso, criei as classes Detection, que representa cada bounding box, calcula o seu centro e em quantos frames ela apareceu, e DetectionTracker, que armazena as detecções, faz o “match” entre as BB de um frame e seu antecessor, atualiza as informações auxiliares, como quantos frames tal Detection apareceu, quais são as posições atuais, quantas detecções já foram feitas, e, para o programa não ficar muito pesado, deleta as detecções salvas que não apareceram por 20 frames seguidos.

O funcionamento geral do programa é:

Processa um frame, detecta as bounding boxes e verifica se elas já são existentes ou novas. Atualiza o número de frames que elas apareceram, e considera cada uma que apareceu ao menos 15 frames como uma detecção válida, atualizando o contador. Busca detecções que não apareceram por 20 frames seguidos e as deleta. Atualiza a imagem com as bounding boxes renovadas.

A detecção inicial das BB foi realizada assim como ensinado em sala de aula, foi usado a função **createBackgroundSubtractorMOG2** para separar o fundo dos objetos, em seguida uma máscara foi criada, e a função **cv.findContours** foi aplicada na mesma. Todos os contornos com área maior que um certo limiar foram considerados válidos. O tamanho deste limiar depende do vídeo, uma vez que eles têm zooms diferentes.

Apenas usar o background subtractor não foi o suficiente. As câmeras apresentam um pouco de movimento entre os frames, o que acarreta que objetos do fundo que deveriam ser estacionários também são detectados, como muretas nas bordas da estrada e as linhas da estrada. Como pode ser visto na imagem a seguir:

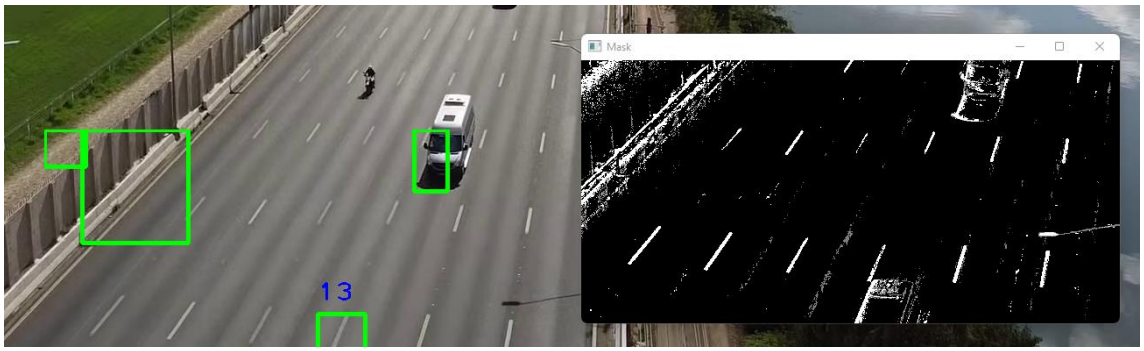

Imagem 1

Um outro problema comum que pode ser observado tanto na imagem anterior quanto na seguinte, é que a parte interior dos veículos não estão sendo detectadas. Como o contorno também não fecha, ao rodar a função de detectar contorno, ou o objeto não é detectado, como na imagem a seguir. Ou em veículos maiores, como na imagem acima, apenas parte do veículo é detectado.

Imagem 2

Para reduzir o primeiro problema, apliquei GaussianBlur na imagem, assim os granulados são reduzidos consideravelmente na máscara:

Imagem 3

Para eliminar mais os ruídos, e ajudar no preenchimento dos objetos, utilizei as funções morfológicas de abertura, fechamento, erosão e dilatação.

Como cada vídeo possui características únicas, a escolha do uso, a intensidade (valores dos kernels) de cada operação morfológica e os valores dos parâmetros da função **createBackgroundSubtractorMOG2** foram ajustados de maneira empírica para cada um dos vídeos.

Como resultado, no vídeo da motociata, por exemplo, vemos as máscaras mais grossas, porém mais preenchidas:

Imagem 4

Um efeito colateral é que dois veículos muito juntos, podem ser detectados como um só.

Execução

Para execução, o programa deve estar no mesmo diretório da pasta vídeos, que contém os 6 vídeos testados. A execução do programa é dada executando na linha de comando:

python ep2 #

Sendo # o índice do vídeo que será analisado:

- 0: motociata
- 1: road_traffic_1
- 2: road_traffic_2
- 3a: road_traffic_3 pista esquerda
- 3b: road_traffic_3 pista direita
- 4: road_traffic_4
- 5a: road_traffic_5 pista esquerda
- 5b: road_traffic_5 pista direita

Os vídeos devem estar com o nome exato dos arquivos fornecidos no e-disciplina.

Observações e Dificuldades

Cada vídeo apresentou suas particularidades, no vídeo da motociata, por algum motivo que ainda não compreendo, veículos pretos não foram detectados muito bem. Tentei até aplicar técnicas de equalização na imagem, para ver se ajudava, mas não adiantou.

A dificuldade do vídeo road_traffic_1 foi que as câmeras mexiam de vez em quando, atrapalhando separar fundos de objetos. Com os tratamentos, acredito que consegui eliminar todos os falsos positivos e contar todos os objetos corretamente.

Ironicamente, no vídeo road_traffic_2, o problema foi inverso. A câmera aparentemente estava posicionada em um semáforo e quando o sinal ficou vermelho os veículos começavam a se tornar parte do fundo, pois estavam parados. Na imagem abaixo, a máscara do carro circulado foi gradativamente reduzindo até sumir completamente.

Imagem 6

Outro efeito curioso que aconteceu está representado na imagem abaixo. No carro à direita, o algoritmo considera apenas o retângulo preto como um objeto, não toda a máscara. No lado esquerdo, pode-se observar a máscara de um carro completamente preenchida. Isso aconteceu, porque automóveis com maior velocidade foram detectados muito mais facilmente pelo algoritmo.

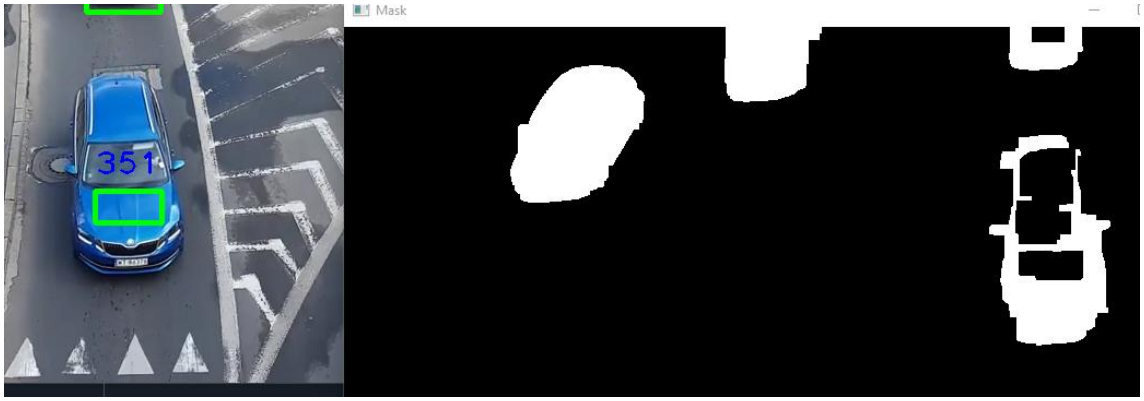

Imagem 7

Como o vídeo road_traffic_3 tem duas pistas com comportamentos distintos, resolvi fazer detectores diferentes, para cada uma delas. O maior problema geral do vídeo foi caminhões e ônibus. Devido ao seu tamanho, diversas vezes são confundidos com vários objetos e, ao mesmo tempo, se fundem com veículos ao seu lado. Na pista esquerda, há momentos em que há congestionamentos e os carros param, causando o mesmo problema citado no vídeo anterior.

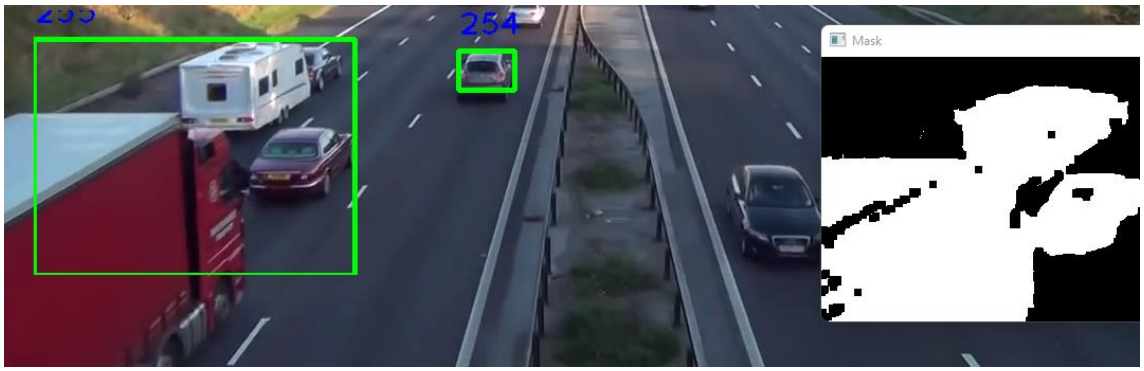

Imagem 8

No vídeo 4, os para-brisas dos carros não estavam sendo reconhecidos. Fazendo que o bounding box deles estivessem parecidos com o da Imagem 7. Apliquei operação de morfologia de fechamento com um kernel bem grande para fechar todo o desenho dos carros. Funcionou na maioria das imagens, mas algumas ainda ficaram com o efeito:

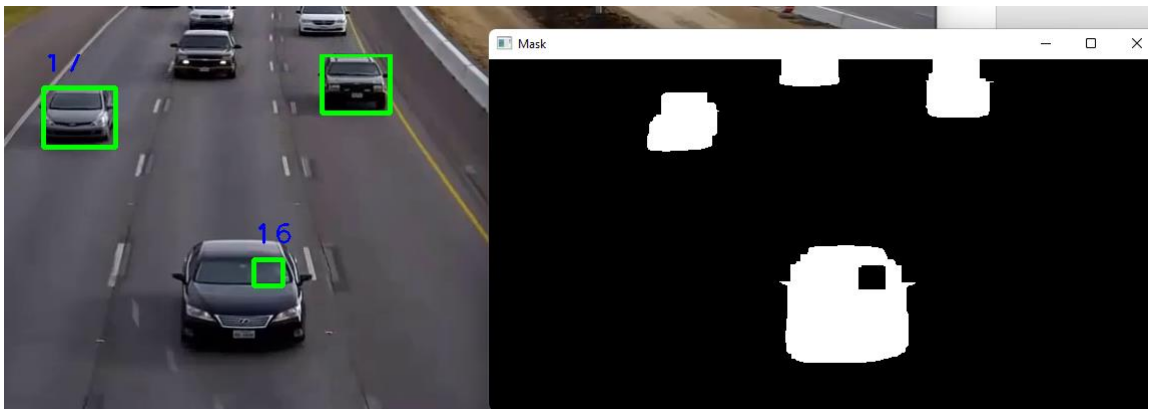

Imagem 9

O processamento do vídeo 5 foi muito tranquilo, os carros estavam em boa velocidade, bem separados e a câmera estava muito estável, por isso não precisei ajeitar parâmetros e transformações morfológicas no mesmo.

Total de veículos contados:

- Motociata: 3531
- road_traffic_1: 23
- road_traffic_2: 423
- road_traffic_3a (lado esquerdo): 2857
- road_traffic_3b (lado direito): 2543
- road_traffic_4 (lado esquerdo): 109
- road_traffic_5a (lado esquerdo): 143
- road_traffic_5a (lado direito):

Conclusão

A tarefa de monitoramento e contagem de veículos é simples de construir, porém difícil de aperfeiçoar, sua qualidade depende muito de fatores externos como posicionamento da câmera e velocidade dos veículos. Resultados superiores poderiam ser obtidos como um algoritmo mais robusto de detecção, como a rede neural YOLO, por exemplo, mas como a proposta da disciplina é resolver com Opencv, preferi não tentar essa solução.