Report of HW3

Huanyu Wang 522030910212

Task 1

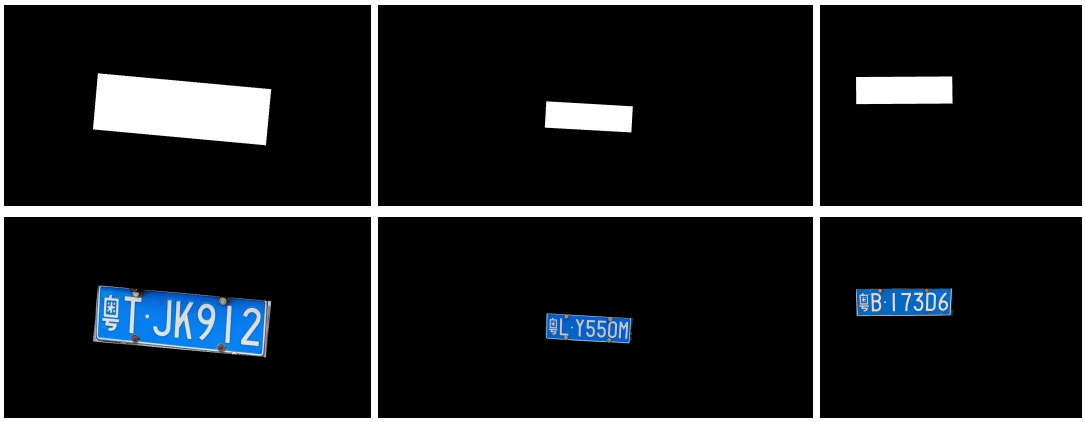
1. 基于第一次作业中我使用的方法，我在此进行优化，最终方法为：使用 HSV 选取对应车牌的特殊蓝色区域，作为 mask 基础，后进行形态学 MORPH_OPEN 和 dilate 操作，最终使用 findContours 选取掩模的外层轮廓，使用 minAreaRect 得到包络旋转举行，并根据长宽比和面积筛选最终的车牌区域。

```
mask_blue = cv2.inRange(hsv, (100, 180, 170), (130, 255, 255))
mask = cv2.morphologyEx(mask_blue, cv2.MORPH_OPEN, kernel,
iterations=2)
mask = cv2.dilate(mask, kernel, iterations=2)
contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
for cnt in contours:
    rect = cv2.minAreaRect(cnt)
    ratio = max(w, h) / min(w, h)
    if 2 < ratio < 6:
        "Details can be found in code_1.py"
```

2. 选取上述方法主要基于以下几个原因：

   - 车牌颜色较为固定 (在此我只对蓝牌进行截取，其余颜色车牌方法类似)
   - 车牌与车身颜色不会完全一致 (可以都是蓝色，可以有同色部分，但是由于车身具有几何形状，打光后的颜色会有渐变，不像车牌作为一个平面，色泽不太会随位置变化)
   - 车牌长宽比较为固定 (假设拍摄图片中车牌面积占整体图片面积比例也较为固定) 我也尝试加入 canny 算子进行边缘检测，以及使用 KMeans 进行聚类，但是效果都没有上述方法好，因此最终选择如上流程。

3. 以下是三张车牌图提取到的 mask 和应用 mask 后的车牌部分图：



4. 以下是具体 LP1.jpg 整体处理的流程，从左往右依次为：蓝色部分 mask，MORPH_OPEN 后，dilate 后，以及最终结果。由此可以看出，最终根据包络矩形的面积和长宽比来筛选车牌的步骤，非常关键，可以剔除很多噪声！

5. 总结：**我认为**上述方法得到的最终结果效果非常好，我甚至感觉我自己手工制作真实 mask 不会比这样得到的 mask 更精确多少，因此交并比就不计算了，完全胜利！

Task 2

1. 选取 Faster R-CNN with ResNet-50 FPN 与训练模型作为基模型

```
model = fasterrcnn_resnet50_fpn(num_classes=2)  # heart + background
```

该模型结构包含以下关键模块：

- Backbone 主干网络：ResNet-50，负责提取图像高层语义特征
- FPN（Feature Pyramid Network）：增强多尺度特征表达能力，提升小目标与大目标检测效果
- RPN（Region Proposal Network）：生成高质量的候选区域
- ROIHead：基于 proposal 进行分类和边界框回归，最终输出检测框

选取此模型的主要原因有：

- Faster R-CNN 是二阶段目标检测方法，相比 YOLO 等一阶段方法在精度上表现更好
- ResNet-50 是经典的深层卷积神经网络，配合 FPN 能有效捕捉多尺度特征
- 本任务中仅需检测一种目标类别，目标数目少，Faster R-CNN 的计算资源可接受，性能稳定

2. 训练细节如下：

- GPU: RTX4090 24G
- 训练参数：

```
epochs = 10
lr = 0.003
batch_size = 4
optimizer = torch.optim.SGD
momentum = 0.9
weight_decay = 5e-4
```

- 训练用时：~17s per epoch

3. 最终在 test 集上的评估结果如下：

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ]
= 0.716
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ]
= 1.000
```

```
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ]
= 0.930
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ]
= -1.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ]
= 0.642
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ]
= 0.751
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ]
= 0.769
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ]
= 0.769
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ]
= 0.769
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ]
= -1.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ]
= 0.672
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ]
= 0.802
```

可以看出，模型的 mAP = 71.6%, AP50 = 100%，展现出很好的目标检测准确率。

4. 以下是训练 epoch_loss 曲线，以及从 test 集中任意选取的三张图片的模型检测结果：