计算机视觉实验报告

目录

摘要

本文档为计算机视觉实验报告，其中具体实现了关于暗通道去雾算法，使用yolov7和face-recognition实现人脸识别，以及关于夜间去雾问题作为实验内容。其中暗通道去雾算法能够有效完成去雾，但是部分物体边缘仍会出现白色模糊带。而人脸识别部分，训练得到的yolov7模型和face-recognition均可以达到很高的准确率和召回率，仅在少数样本中无法准确识别。最后对于夜间去雾模型，主要分析了核心的方法，并就其中的半监督训练部分进行了思考改进。源码附在woodenchair/cv_hw (github.com)。

一、暗通道去雾算法

1.1 方法介绍

文章提出的暗通道去雾算法[1]是一种基于统计学习的方法，用于改善单幅图像在雾霾条件下的视觉效果。该算法首先利用暗通道先验，观察到在无雾图像中大多数局部区域至少在一个颜色通道内包含低强度像素。通过这一观察，结合雾天成像模型，算法能够有效估计出图像中各点的雾霾密度。

此时图像已经基本完成了去雾这样一个部分，但是生成的图片可能会产生伪影，进一步，使用软抠图技术对传输图进行细化，以减少伪影并恢复清晰的无雾图像。此外，算法还能自动估计大气光，从而进一步提升去雾效果。这种方法在多种雾天图像上验证了其有效性，能够显著提高场景的能见度和色彩保真度。

1.2 实验结果

下面选取了部分图片进行展示，其中左图为原图，右图为去雾后的图像。

图一(a):去雾前

图一(b):去雾后

图二(a):去雾前

图二(b):去雾后

图三(a) :去雾前

图三(b) :去雾后

图四(a) :去雾前

图四(b) :去雾后

1.3 结果分析

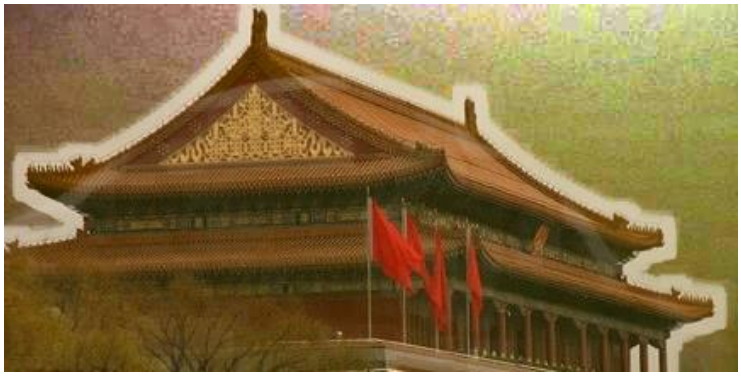
上面是算法实现后的最终效果，可以看到雾已经被有效地去除了，同时我研究上图的图一的去雾前后的时候，主观感觉去雾后的图片和去雾前的深度感知完全不同，考虑论文中给出的雾天成像模型，显示是大气光去除过多所导致的，论文中是通过在求解大气光时设置修正系数ω来优化视觉效果的，上面的图片修正系数ω为0.95，下面展示了omega取值为0.6的情况，主观感受上来看，ω=0.6时，图片的纵深更加明显。

图五(a): ω=0.95

图五(b): ω=0.6

但是对于去雾处理之后得到的图片，在非白色物体和白色背景相交的边缘会出现伪影，如下图图六(b)所示，论文中所给的方法是使用soft matting来解决这个问题，但是实际实现中，发现其中求解laplacian矩阵以及后续方程的求解，算力要求过大，并且效果并不理想，噪声影响严重，后来查看何凯明博士研究组的后续研究进展，发现使用引导滤波能够有效解决伪影的问题，并且实现简便并且具有更好的效果，因此之后选择放弃使用soft matting而是用引导滤波的方法解决伪影的问题。

图六(a):原图



图六(b):处理后有伪影

图六(c):去伪影后

1.4 思考拓展

1.4.1图像深度信息

对于去雾所导致的图像深度信息的缺失，论文所给出的方法是使用修正系数ω来恢复图像深度信息，能使图像具有更好的视觉效果。所以，对于深度信息恢复这个不分，我觉得如果为了提升图像的视觉效果，恢复深度信息，可以考虑使用知识蒸馏，通过使用完善的美学模型作为教师模型，单独为图像深度信息恢复设置学生模型，使用知识蒸馏的方式帮助图像恢复深度信息

1.4.2物体边缘

观察图七的去雾效果图，我们可以发现，当对整体物块进行去雾时，去雾效果很好，但是，对于与白色背景相交的物体边缘部分，是由伪影去除后的白色模糊带所组成的，考虑实际实现的算法，应该是由去雾时处理的patch_size设置的大小所决定的，论文中所给的合适的patch_size为15*15，因此对于物体边缘可能会出现白色模糊带这样，可以使用PatchMatch算法来对图像进行处理，这样对于图像处理之后，能够使得

图七(a)

图七(b)

1.5关键源码

最后在这里附上去雾算法的关键代码

1.5.1求解大气光强

```
for y in range(height):
            for x in range(width):
                window_size = self.patch_size // 2
                # 提取局部窗口
                x_start = max(0, x - window_size)
                x_end = min(width - window_size, x + window_size)
                y_start = max(0, y - window_size)
                y_end = min(height - window_size, y + window_size)

                local_window = self.image[y_start:y_end + 1, x_start:x_end + 1
]
                r_ratio = local_window[:, :, 0] / result['R']
                g_ratio = local_window[:, :, 1] / result['G']
                b_ratio = local_window[:, :, 2] / result['B']

                min_ratios = np.min([r_ratio, g_ratio, b_ratio], axis=0)
                local_min_ratio = np.min(min_ratios)
                # 计算 t[k]
                self.t_values[y, x] = 1.0 - local_min_ratio * self.omega
```

1.5.2引导滤波

```
def guided_filter(self):
    r = 20
    d = cv2.cvtColor(self.image, cv2.COLOR_BGR2GRAY).astype('float64') / 255
    # i = self.image
    t = self.t_values
    eps = 1e-4
    # 1
    mean_I = cv2.blur(d , (r, r))
    mean_p = cv2.blur(t , (r, r))
    corr_I = cv2.blur(d * d , (r, r))
    corr_Ip = cv2.blur(d * t , (r, r))
    # 2
    var_I = corr_I - mean_I * mean_I
    cov_Ip = corr_Ip - mean_I * mean_p
    # 3
    a = cov_Ip / (var_I + eps)
    b = mean_p - a * mean_I
    # 4
    mean_a = cv2.blur(a , (r, r))
    mean_b = cv2.blur(b , (r, r))
    # 5
    q = mean_a * d + mean_b
    q = cv2.max(q, 0.25)
```

```
24.     self.t_values = q
```

二、人脸识别

2.1 数据集和方法介绍

关于第二题，我选择的是人脸检测这样一个问题，并且由于算力的原因，在这里我只选择了yolov7模型[2]进行了训练。另外一种实现则是使用face-recognition所提供的人脸检测。

其中选择的人脸数据集来自CelebA数据集。其中CelebA一共含有202599张图片，训练全部的数据时间耗费过长，并且根据后续实际训练的效果来看，单纯提高数据量也没有办法有效提高模型性能，因此仅选取了10329张图片作为数据集。其中训练集、测试集与验证集的比例约为8:1:1。

来自于CelebA的数据无法直接用于yolov5训练，我在这里先将数据集的label转换为yolo类型的标签，需要将所检测的人脸的宽w，高h，图片中心点的坐标x，y进行中心化处理。

对于模型训练所用到的参数，在这里，我修改了workers设置为8，batch_size为8，epoch本来设置是50，但是实际上模型训练到大约10个epoch左右时，性能已经基本确定，最后，在30个epoch左右完成训练。训练使用了colab所提供的云GPU进行训练，在配置好数据集等相关部分后，运行以下代码开始模型训练。

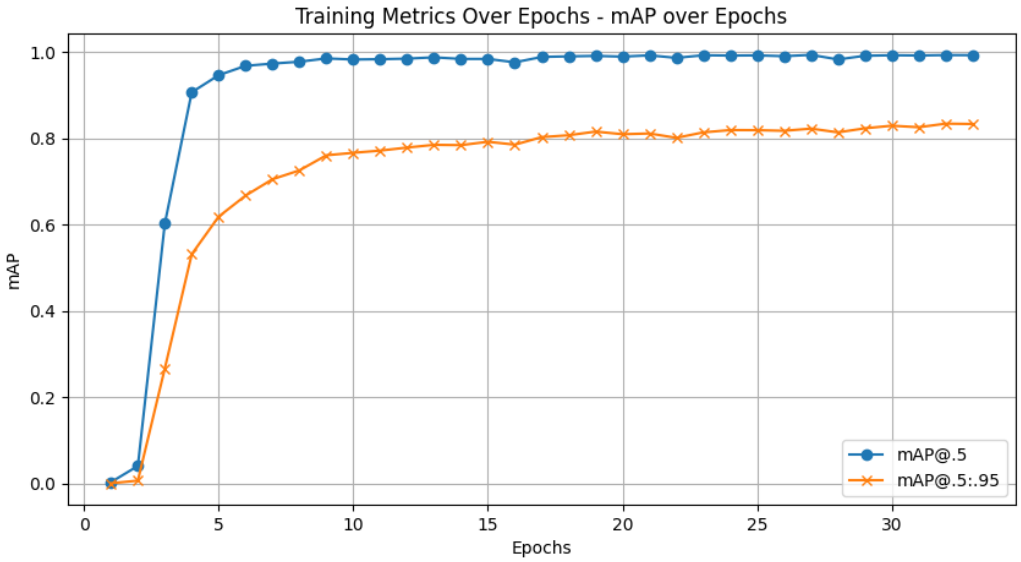
```
!python train.py --workers 8 --device 0 --batch 16 --data ./celeba/celeba.yaml --img 640 --cfg ./cfg/training/yolov7.yaml --weights '' --project /content/drive/MyDrive/yolov7_train --name yolov7 --epochs 50 --resume ../drive/MyDrive/yolov7_train/yolov75/weights/last.pt
```

而对于face-recognition中的人脸检测的实现，具体使用的是使用 dlib 的 HOG (Histogram of Oriented Gradients) 算法来检测图像中的人脸。能够完成单个图像中多人脸的检测，简单易用。

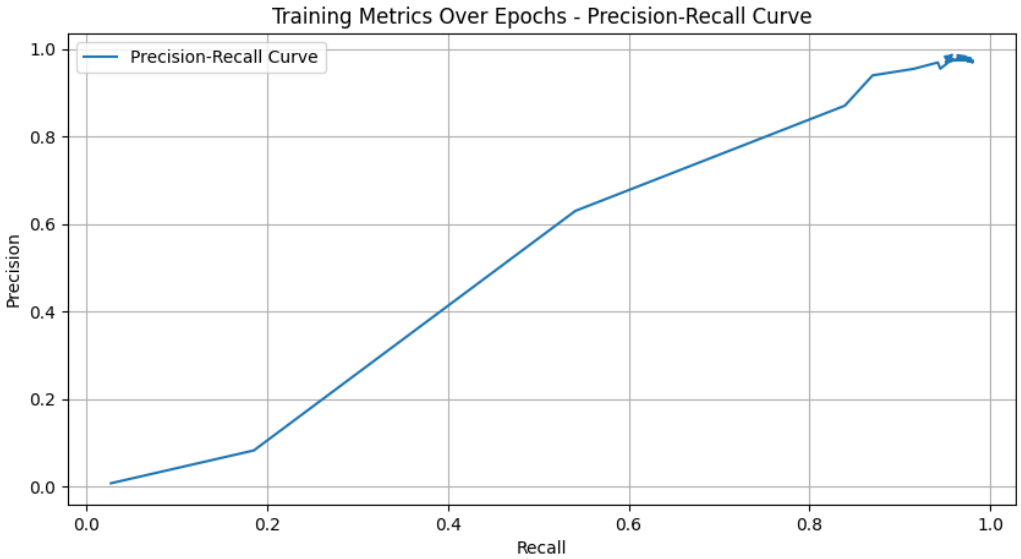
2.2 训练结果

Yolov7 训练时所用的所用的训练指标被储存到了 result.txt 文件夹中，我选择了其中的四个指标结果并进行了可视化，图八是反应 mAP(mean average precision)随 IoU(intersection over union)的变化，具体对于图八中的 mAP_.5 和 mAP_.5:.95，其中 mAP_.5 代表 Iou 为 0.5 时每一类的所有图片的 AP 值，然后所有类别求平均。而 mAP_.5:.95 则分别代表表示在不同 IoU 阈值（从 0.5 到 0.95，步长 0.05）（0.5、0.55、

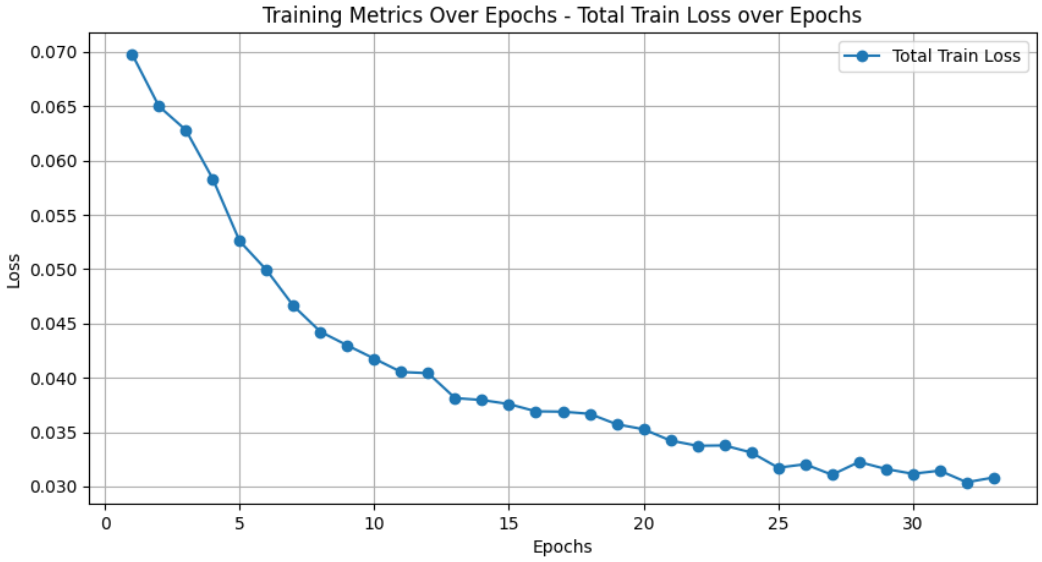
0.6、0.65、0.7、0.75、0.8、0.85、0.9、0.95）上的平均 mAP。图九是训练过程中的 PR 曲线。图十和图十一反应的是训练过程中的不同类型损失的变化。



图八:不同Iou下的mAP值变化



图九:训练过程中precision和recall的数值对应联系



图十：训练矩阵的train loss变化

图十一：训练矩阵的box loss变化

2.3 性能对比

由于在训练过程中，在这里，我从查准率，召回率，处理速率对模型的性能，进行了考虑，而face-recognition中，由于数据集中提供的是人脸框的数据，于是我使用IoU值(intersection of union)来判断是否准备检测到人脸的,其中临界值为0.8，如果超过该临界值，则视为准确检测，否则认为检测失败。

在此基础上，选取了测试集中的一部分数据进行测试，其中yolov7查准率和召回率分别达到了96.35%和97.06%，face-recognition的查准率和召回率分别为97.56%和98.12%，两个模型的在查准率和召回率中相差不大，但是两个模型的识别速度相差较大，face-recognition的识别时间差别很大，检测所需要花费的时间2-200s不等，如果，人脸出现

遮挡物时会花费很长时间，yolov7的识别速率较快，平均每秒可以处理8张图像。

显然综合来看，yolov7的性能要优于face-recognition，但是face-recognition可以视为一个轻量化的模型，已经被完好的封装，性能上在预想中确实会和yolov7有一定差异。

图十二(a)：人脸图片

图十二(b)：识别到的人脸

图十三(a)：人脸图片

图十三(b)：识别到的人脸

2.4 模型不足

对于yolov7模型，在测试的时候发现，对于部分有遮挡的人脸，其中遮挡可能包含有多种方式，比如说光线所导致的阴影，无法正常识别，以及各种障碍物的遮挡可能会导致无法正常检测到人脸。而且，根据后续测试发现，face-recognition中基于dlib实现的人脸检测并不能很好地检测小孩的人脸，

图十四(a)

图十四(b)

图十五(a)

图十五(b)

三、夜间去雾

3.1 领域简介

对于最后一个部分我选择的是一个解决夜间去雾问题，作者设计了夜间去雾的baseline SFSNiD(Semi-supervised Nighttime Dehazing)[3]。夜间图像去雾与白天图像去雾相比，面临一些特殊的问题，其中夜间场景中多种光源和雾、噪声之间的可能会产生耦合。夜间场景可能存在多个活跃的彩色光源，这些光源的亮度较低，可能导致局部化的雾、光晕和噪声，这些干扰在图像中混合且具有不一致的频率特性，会大大增加夜间去雾的难度。

对于一个夜间其中夜间去雾相比传统去雾具有更多的问题，大气光强度在夜晚相比于白天更低，并且在夜间，光源往往不单一，图像会受到比如路灯、车灯等其他光源的影响，会导致自然和非自然的，并且夜间场景中的雾气和由强光源引起的光晕现象往往是耦合的，这使得它们难以分离，如下图所示。

图十六(a):

图十六(b):

图十七(a):含雾图片

图十七(b)实际图片

3.2 方法分析

而对于上面所提到的问题，论文中所使用到的主要的技术可概括为下面这三个部分。其中第三个是关于半监督训练策略的，会在3.3详细介绍。

1) 空间-频率域信息交互模块(SFII)：该模块利用空间注意力和频率谱滤波来处理上述第一个问题，即同时处理具有局部化、耦合和频率不一致特性的雾、光晕和噪声。

2)局部窗口基础的亮度损失：设计了一种局部窗口基础的亮度损失函数，用于在半监督训练过程中抑制雾和光晕，同时实现真实的亮度效果。

3.2.1 SFII

对于第一个模块，在整个去雾模型中，SFII是用于增强夜间图像去雾的性能，其

核心原理在于融合空间注意力和频率谱滤波技术，以同时处理图像中的雾、光晕和噪声等失真现象。

其中空间注意力机制专注于捕捉图像中的关键区域，例如物体的边缘和纹理，通过学习到的注意力权重加强这些区域的处理。频率谱滤波通过将图像转换到频率域并应用滤波器，有效去除由雾引起的高频损失和噪声。并且SFII模块采用多尺度处理，使网络能够综合考虑图像的局部细节和全局上下文，从而全面理解和恢复图像。

而对于图像内容的变化，动态频率谱聚合利用动态滤波器对不同频率通道的信息进行自适应聚合，从而对此做出反应。并且SFII通过双域交互，能够整合空间域和频率域的信息，可以在此基础上加强特征表示，使网络能够更有效地学习并恢复图像特征。除此之外，SFII还包含局部感知和非线性映射，这允许网络在空间域内处理局部区域，并通过非线性映射进一步丰富特征表示，以捕捉图像中的复杂特性。

3.2.2 亮度损失函数

将图像划分为不重叠的局部窗口，并为每个窗口计算一个亮度值。高亮度区域可能对应于活跃的光源或靠近光源的物体，而低亮度区域可能对应于远离光源的物体和背景。基于对夜间图像亮度的统计分析，我们可以假设去雾后的图像亮度应该低于雾图像的亮度，并且在实际中，这也与成像模型一致。

论文中并没有具体给出一个具体的损失函数，但是描述了一个具有单调递增特性的幂函数，用于处理局部窗口的亮度值。这个幂函数的一般形式可以表示为：

$$Activation\ Function(f(x)) = f(x)_{\kappa}$$

其中κ是一个大于1的超参数。损失函数确保了去雾图像的亮度不仅整体上接近真实情况，而且保持了高亮度和低亮度区域之间的相对数值关系。

3.3 思路改进

在所使用的方法中，论文考虑到真实世界数据和模拟数据之间域不一致的问题，于是采用了基于伪标签的重训练策略。这涉及到使用在模拟数据集上预训练的模型来生成真实世界数据的伪标签，并将这些伪标签与真实标签结合用于进一步训练。从而帮助模型更好地适应真实世界数据的分布。实际上，这个部分最关键的点在于帮助模型更好地适应真实世界的数据分布，基于这个点，我们可以考虑一下几个方面去解决。

3.3.1伪标签生成

在考虑使用更加复杂的伪标签生成方法时，我们可以通过引入置信度评分来提高伪标签的质量。核心部分在于利用模型的预测概率来计算每个预测结果的置信度。

例如，我们可以采用模型输出的类别概率的最大值作为初步置信度指标。进一步地，通过计算预测概率分布的熵，我们可以量化预测的不确定性，从而得到更为精细的置信度评分。

在计算得到置信度评分之后，可以过滤掉那些置信度低的预测结果，以减少噪声对模型训练的影响。同时，当已有的伪标签数量达到设置的超参数阈值之后，我们可以对标签进行筛选，挑选出置信度最高部分的预测作为伪标签，这些标签具有更高的可信度，可以更有效地指导模型学习。此外，基于置信度评分，我们可以自适应地调整模型的学习率，对于高置信度的伪标签使用较大的学习率，而对于低置信度的伪标签则使用较小的学习率或者完全不使用这些标签。从而有效提高模型在真实数据下的泛化性。

3.3.2 训练策略

优化训练过程也是一个可以考虑的方向，我们可以考虑尝试不同的重训练策略，比如渐进式重训练，即逐步引入真实数据的标签信息。首先将渐进式训练描述为一个分阶段的过程。初始阶段，模型仅使用少量的标记数据进行训练。随着训练的进行，逐渐将通过模型自身预测得到的高置信度伪标签引入到训练集中。这个过程可以通过设置一个置信度阈值来实现，只有当模型对其预测结果足够自信时，这些预测才会被视为训练数据。这其中的置信度的实现可以采用和3.3.2中所用到的类似的方法实现。

在实现时，首先，模型在少量标记数据上进行预训练。然后，对于未标记的数据集，模型产生预测，并计算每个预测的不确定性（例如，通过预测概率的逆或熵来衡量）。选择那些不确定性低于某个阈值的样本作为伪标签，并将其加入到训练集中。随着时间的推移，逐渐增加这些伪标签样本的比例，同时继续优化模型。

参考文献

[1] Y. Xu, X. Guo, H. Wang, F. Zhao and L. Peng, "Single image haze removal using light and dark channel prior," 2016 IEEE/CIC International Conference on Communications in China (ICCC), Chengdu, China, 2016, pp. 1-6, doi: 10.1109/ICCChina.2016.7636813.

[2] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023.

[3] Cong, Xiaofeng, et al. "A Semi-supervised Nighttime Dehazing Baseline with Spatial-Frequency Aware and Realistic Brightness Constraint." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.