

恶劣环境下图像算法数据增强方法

刘洪宇, 杨林⁺, 姜蕾

(中国航天科工集团第二研究院七〇六所, 北京 100854)

摘要:为解决恶劣环境下数据采集难度较大、数据匮乏导致模型性能受限的问题,提出一种基于风格迁移的数据增强方法,用于增加恶劣环境下的样本数量,提升模型在恶劣环境下的鲁棒性。建立包含22 500张图片的数据集,使用卷积神经网络进行图片去重,进行手工标注,用于进行分类模型的训练。设计正常环境和恶劣环境对比实验,验证提出的数据增强方法效果,实验结果表明,该方法可以有效提升分类模型在恶劣环境下的鲁棒性。

关键词:恶劣环境;人工智能;图像算法;数据增强;风格迁移

中图分类号: TP399 **文献标识码:** A **文章编号:** 1000-7024 (2021) 09-2545-07

doi: 10.16208/j.issn1000-7024.2021.09.019

Data augmentation in harsh environments for computer vision algorithms

LIU Hong-yu, YANG Lin⁺, JIANG Lei

(Institute 706, Second Academy of China Aerospace Science and Industry Corporation, Beijing 100854, China)

Abstract: In some harsh environments, data acquisition is difficult, and the lack of data has a great impact on model performance. A data augmentation method based on style transfer was proposed to increase the number of samples in harsh environments and improve the robustness of the model. The established dataset contained 22 500 manually-labeled images, where a convolutional neural network was used for image deduplication. The classification model was trained on this dataset to verify the proposed data augmentation method. The contrast experiment between normal environment and harsh environment was designed to verify the effect of the proposed method. Experimental results show that this method can effectively improve the robustness of the classification model in harsh environments.

Key words: harsh environments; artificial intelligence; image algorithm; data augmentation; style transfer

0 引言

在计算机视觉领域,随着优化方法^[1,2]的进步和公开数据规模的提升,基于深度学习的目标识别^[3,4]、目标检测^[5,6]、人脸识别^[7,8]等算法效果已取得较大提升。这些算法受潮湿、震动、电磁干扰等外部恶劣环境的影响较小,只要保证输入图像质量,算法即可稳定运行。但风沙、大雾等环境会为捕捉到的图像带来较大噪声,本文实验结果表明,若不进行处理,这些恶劣环境引入的噪声将对模型效果产生较大影响。

传统数据增强方法可解决大部分恶劣环境下出现概率较高的图像问题,例如通过对亮度、对比度和饱和度的调整,模拟逆光、过度曝光、低亮度等情况;通过图像掩码

模拟物体遮挡情况等,但传统方法无法模拟大雾、风沙等恶劣环境,为解决这一问题,提出了一种基于图像风格迁移的数据增强方法,主要工作有以下几点:

(1) 将图像风格迁移技术用于数据增强,对训练集中的图片进行大雾、风沙环境下的数据增强,增加恶劣环境下的样本数量。

(2) 使用 PeleeNet^[9]作为分类网络骨架,对激活函数、损失函数以及网络训练方法进行优化调整,并使用改进后的网络进行图像分类任务,验证提出的数据增强方对恶劣环境下图像分类模型鲁棒性的提升。

(3) 建立了包含22 500图片的数据集,包括完整的图片爬取、图片去重以及人工标注流程处理,用于模型的训练和测试。

收稿日期: 2020-12-07; 修订日期: 2021-05-18

作者简介: 刘洪宇 (1993-), 男, 河北廊坊人, 硕士, 工程师, 研究方向为计算机视觉; ⁺通讯作者: 杨林 (1989-), 男, 北京人, 博士, 高级工程师, CCF 专业会员, 研究方向为人工智能、嵌入式计算; 姜蕾 (1993-), 女, 山东烟台人, 硕士, 工程师, 研究方向为人工智能与机器学习。E-mail: 190978752@qq.com

1 基于风格迁移的数据增强

由于恶劣环境下数据采集难度大、数据标注成本高等原因,数据集规模较小是一个较为严重的问题。这将导致模型拟合的数据分布严重偏离真实分布,影响模型的泛化能力。为解决这一问题,在进行模型训练时,可用数据增强技术对已有数据进行扩充,在增加训练样本量的同时,模拟模型需要处理的各种场景,提升模型在这些场景下的鲁棒性。传统数据增强技术常采取如下几种方法:像素级别的数据增强、像素块级别的数据增强、多张图片叠加进行数据增强。

像素级别的数据增强技术通过修改原始图像中的某些像素值来增加样本多样性。如对图像亮度、对比度、色调、

饱和度的调整,在原始图像上叠加随机噪声,或对图像进行随机缩放、裁剪、翻转、旋转等操作。像素块级别的数据增强技术通过对原始图像中某些区域的像素信息进行修改来达到样本扩充的目的。如 Zhun Zhong 等^[10]提出使用随机擦除方法进行数据增强,该方法从原始图像中随机选取一块矩形区域,对该区域使用随机值或零进行填充,模拟物体被遮挡的场景。网格掩码^[11]方法则从原始图像中随机选取多个像素块,将所有选取的像素块进行零填充。此外,一些研究人员通过一次使用多张图片叠加的方式进行数据增强。例如 MixUp^[12]方法中,将两个随机采样的样本进行叠加来构造新样本。图 1 中展示了传统数据增强方法的效果,其中图 1(a)为原始图片,其余为应用不同数据增强方法得到的图片。

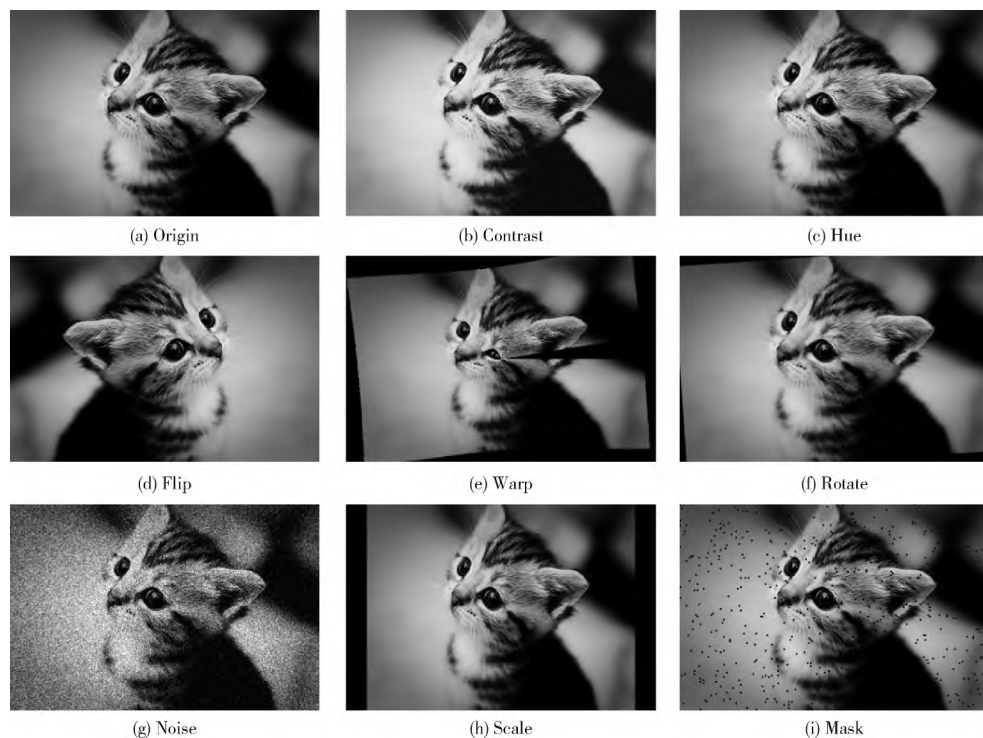


图 1 传统数据增强方法效果

为缓解传统方法无法模拟的大雾、风沙等恶劣环境对图像算法产生的影响,可利用图像风格迁移技术进行数据增强。图像风格迁移分为真实图像风格迁移^[13]和艺术图像风格迁移^[14]两类,真实图像风格迁移算法根据参照图片的风格来改变目标图片风格,同时保证输出图片的真实性。这类算法可用于不同季节的图像风格转换、白天向夜间图像转换等。而艺术图像风格迁移则主要用于将真实图像转换成绘画风格的图像。图 2 展示了这两类风格迁移算法的效果,通过图 2 中上下两行图片转换效果的对比可以看出,这两类算法虽然都能起到风格转换的作用,但转换后图像的真实性存在较大差异。

本文中使用的真实图像风格迁移算法为训练样本添加恶劣环境下图像的风格,来增加恶劣环境下样本数量。算法分为两个步骤:风格转换和图像平滑,风格转换将目标图片向风格图片进行变换,针对变换后的图像语义相同的部分风格不连贯的问题,使用图像平滑方法消除这种影响,使变换后的图像风格更加接近真实图像。

图 2 中,图 2(a)列为风格图片,图 2(b)列为目标图片,图 2(c)列为风格迁移后的图片。第一行为真实图像风格迁移效果,第二行为艺术图像风格迁移效果。

1.1 风格转换

本文中图像的风格转换借助卷积神经网络来完成,在

进行转换之前, 首先训练出一个用于图像重建的自编码器。自编码器中, 使用 VGG-19^[15] 作为编码器 ϵ , 固定其参数, 并训练与之结构对称的解码器 D 进行输入图像重建。

VGG-19 中使用最大池化进行下采样, 在解码器中, 为了保留特征图的空间信息, 利用编码器中的池化掩码进行反池化操作, 达到上采样的目的。自编码器结构如图 3 所示。



图 2 图像风格迁移算法效果

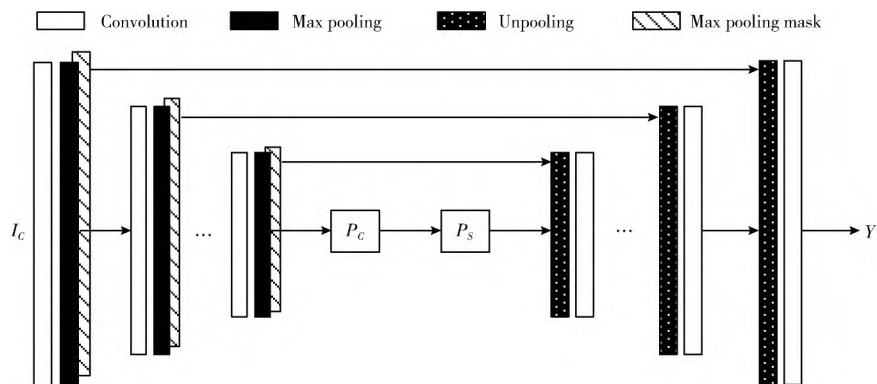


图 3 自编码器结构

转换过程可以分为 3 个步骤:

(1) 使用编码器对风格图片 I_s 和内容图片 I_c 进行特征提取, 得到两张图片对应的特征向量 $H_s = \epsilon(I_s)$ 、 $H_c = \epsilon(I_c)$ 。

(2) 使用如下公式对 H_c 进行转换, 得到转换后的特征向量 H_{cs}

$$H_{cs} = P_s P_c H_c \quad (1)$$

式中: $P_c = E_c \Lambda_c^{-\frac{1}{2}} E_c^T$, $P_s = E_s \Lambda_s^{-\frac{1}{2}} E_s^T$, Λ_s 与 Λ_c 是矩阵 $H_c H_c^T$ 和 $H_s H_s^T$ 对应的特征值组成的对角矩阵, E_c 与 E_s 为相应特征向量对应的正交矩阵。通过上述变换, 可以保证 $H_{cs} H_{cs}^T = H_s H_s^T$ 。

(3) 使用解码器对转换后的特征向量 H_{cs} 进行解码, 得到风格转换后的图片 $Y = D(H_{cs})$ 。

1.2 图像平滑

解码器输出的图像存在过于风格化且风格不连贯的问题, 需要使用图像平滑方法消除这种影响, 使其更加接近

真实图像。图像平滑的优化目的有两点: ①使图像中相邻区域内容相似的像素具有相似的风格; ②使平滑处理前后图像差异较小, 保留风格变换后图像的整体风格。

对于输出图像 Y , 像素之间的相似矩阵可表示为 $W = \{w_{ij}\} \in R^{N \times N}$, 其中 N 为图像 Y 中的像素数量。根据上述优化方向, 将目标函数定义为

$$\operatorname{argmin}_r \frac{1}{2} \left(\sum_{i,j=1}^N w_{ij} \left\| \frac{r_i}{\sqrt{d_{ii}}} - \frac{r_j}{\sqrt{d_{jj}}} \right\|^2 + \lambda \sum_{i=1}^N \|r_i - y_i\|^2 \right) \quad (2)$$

式中: y_i 是图像 Y 上的像素值, r_i 是待求解图像 R 上的像素值, $d_{ii} = \sum_j w_{ij}$ 。为了平衡目标函数中两项定义的大小, 引入 λ 参数, 实验中 λ 取 0.1 可以取得较好的效果。上式可以通过解析解直接求出, 作为图像风格迁移的最终结果。

2 恶劣环境下的图像分类模型

本文通过图像分类任务来验证风格迁移方法对恶劣环

境下模型效果的提升。由于恶劣环境下工作的模型通常部署在嵌入式设备中,因此本文选用 PeleeNet 这一轻量级网络作为基本骨架,并根据近年来网络训练及加速方面的研究成果对其进行相应改进。

2.1 激活函数

Relu 激活函数由于形式简单、不易产生梯度消失与梯度爆炸的问题,在神经网络中被普遍使用,原始的 PeleeNet 网络结构也将其作为网络的激活函数。

神经网络训练时,出于速度和精度上的综合考量,Pytorch、Tensorflow 等深度学习框架默认使用单精度浮点运算在 GPU 上进行模型训练。模型训练结束后,为了获得更快的推理速度、更小的模型尺寸,通常会使用更低的精度将其部署到移动端设备上,如 8 bit 或 16 bit。对于 Relu 激活函数来说,没有对网络激活值进行限制,激活值的输出范围是 $[0, +\infty)$,这将导致模型以低精度方式运行时,较大的激活值可能产生精度损失。因此,本文将激活函数替换成 Relu6^[16],降低量化过程带来的模型精度损失,Relu6 激活函数的数学定义如下

$$y = \min(\max(0, x), 6) \quad (3)$$

式中: x 和 y 分别为激活函数的输入、输出值。

2.2 改进的分类损失

原始 PeleeNet 中,在进行网络训练时,使用交叉熵作为网络的损失函数

$$H(p, q) = - \sum_{i=1}^n p_i \star \log(q_i) \quad (4)$$

式中: p 为 one-hot 编码的样本真实标签, q 为 Softmax 输出的类别置信度。Softmax 公式如下

$$q_i = \frac{e^{f_i}}{\sum_{j=1}^C e^{f_j}} \quad (5)$$

式中: C 为类别数, $f = [f_1, f_2, \dots, f_C]$ 为神经网络输出的特征向量。

本文中,对分类损失函数进行了改进,保留交叉熵损失函数,同时在损失函数中对所有 BatchNorm^[17]层的缩放参数进行了 L_1 正则化。 L_1 正则化会使得缩放参数在训练过程中呈现稀疏化的趋势,这一技术由 Zhuang Liu 等^[18]提出,并被应用于卷积神经网络通道剪枝中。在进行剪枝通道选取时,对缩放参数进行全局排序,将排序后的缩放参数作为通道保留的概率,文中将这一技术称为“稀疏性训练”。在实验中发现,这一技术不仅能起到剪枝通道选取的作用,其作为一种正则化手段,同样能提高网络性能,因此,本文中将其作为损失函数的一部分。加入上述正则化技术后,损失函数计算公式如下

$$L_{cls} = L_{ce} + \lambda \sum_{\gamma \in \Gamma} |\gamma| \quad (6)$$

式中: L_{ce} 为交叉熵损失, Γ 为网络中所有 BatchNorm 层缩放参数 γ 组成的集合,为了平衡两项损失的权重,在 L_1 正

则化损失前加入平衡因子 λ , λ 取值通常为 $1e-4$ 或 $1e-5$ 。

2.3 标签平滑

使用 PeleeNet 进行网络分类时,网络最后一层为全连接层,全连接层输出的神经元个数为分类任务的类别数 C ,这 C 个输出值分别作为每个类别的置信度,表示为 f 。通常会使用 Softmax 对 f 进行标准化操作,保证 $f_i > 0$ 且 $\sum_{i=1}^C f_i = 1$ 。

另一方面,设 y 为图像的真实标签,则使用独热编码后的标签可表示为

$$q_i = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

当使用交叉熵定义分类损失并对其进行优化时,最优解会出现在 f_y 趋于无穷大而其他值尽可能小的情况下,这一优化方向容易导致模型过拟合。为解决这一问题,本文中使用标签平滑技术对模型训练时使用的独热编码进行处理。标签平滑的思想在 Inception-v2^[19]训练时被提出,此后研究人员对这一技术从理论^[20]和实践^[21]层面进行了详尽分析,目前大量实验表明使用标签平滑可以提升模型性能。标签平滑技术对样本的真实标签进行如下编码

$$q_i = \begin{cases} 1 - \epsilon & \text{if } x = y \\ \epsilon / (C - 1) & \text{otherwise} \end{cases} \quad (8)$$

式中: ϵ 是一个较小的常量,通常取 0.1。

3 实验分析

3.1 实验数据集

本文所用数据集为包含 22 500 张图片的自然图像数据集,共 9 个类别,每个类别 2500 张图片。数据集中小部分样本来源于 ImageNet 图像分类数据集,ImageNet 数据集中每个类别包含 1000 张图片,图片已完成标注,本文从中选出所需的 9 个类别,并对数据进行人工筛选。另一部分图片使用类别关键字在搜索引擎中检索,并爬取检索结果。为了保证图像质量,每次只爬取检索到的前 500 张图片,此外,使用多种语言的关键词在搜索引擎中检索。

从搜索引擎爬取的图片存在部分重复,本文借助卷积神经网络对所有爬取到的图片进行去重。具体分为两个步骤:①使用在 ImageNet 上预训练的 MobileNet 模型对数据集中的每张图片进行特征提取;②对提取到的特征采用 1:N 的方式逐一比较,删除相似度较高的图片。

在对图片进行人工标注时,由于在数据爬取时图片按照关键词进行检索,因此标注时只需要按照类别进行“是”或“不是”的判断任务即可。

3.2 图像风格迁移实验

图像风格迁移实验使用 FastPhotoStyle 算法的开源实现进行,软件环境为:Ubuntu 16.04、CUDA 9.1、Anaconda3、PyTorch0.4.0。硬件环境使用 NVIDIA RTX 2080Ti GPU

进行推理加速。

在 PyTorch 深度学习框架中, 对神经网络训练数据进行数据增强时, 由于数据变换速度较快, 只需预定义好数据增强策略, 对数据进行变换在加载数据时与网络训练同时进行即可。但图像风格迁移耗时较多, 无法做到实时处理, 因此本文采用离线数据增强的方式对数据集中样本进行处理。实验中使用了沙尘、大雾、雨天、雪天 4 种风格的图片对数据集中样本进行风格迁移。

内容图片在向沙尘、大雾风格进行迁移时可以输出较好的结果, 而向雨、雪风格迁移后的图片质量较差。由于在向沙尘、大雾风格进行迁移时, 内容图像主要进行了色调上的修改, 而向雨、雪风格进行迁移不仅需要色调变换, 还需要向内容图片中添加雨、雪元素, 导致图像过于风格化。

模型训练时, 批大小设置为 128。实验中使用 Adam^[22] 优化方法进行模型参数更新, 学习率初始值设置为 $3e-3$ 。每次实验设置的迭代次数均为 160 轮次 (epoch), 实验中采用学习率按步长衰减的策略, 每 40 轮迭代, 将学习率调整为原来的 1/10。

此外, 由于数据量的限制, 本文并未从头开始训练随机初始化的模型, 而是根据已发布的 ImageNet 预训练模型进行微调训练。模型微调训练流程如图 4 所示。

在图 4 中, 模型由特征提取器 F 和分类器 CLS 两部分组成。模型微调训练分为 3 个阶段: 模型初始化阶段、第一微调阶段、第二微调阶段。在模型初始化阶段, 除网络最后的全连接层外, 其余层参数均使用预训练模型参数进行初始化, 全连接层使用 Kaiming_normal^[23] 方法进行初始化。在模型初始化的同时, 从预训练模型中加载的参数被冻结, 不参与第一阶段的微调训练。在第一微调阶段, 对全连接层的参数进行更新, 当训练结束后, 将网络中所有参数设置为可更新状态。最后, 在第二微调阶段对模型所有参数进行更新, 得到收敛后的模型。

3.3 图像分类网络改进实验

3.3.1 实验配置

本文在进行网络训练时, 使用的硬件环境为: NVIDIA RTX 2080Ti GPU, Intel® Core™ i7-8700 CPU@3.20 GHz, RAM 32.0 GB。所有算法都是在 Windows 10 操作系统下编程实现, 代码使用 PyTorch1.4 深度学习框架进行编写,

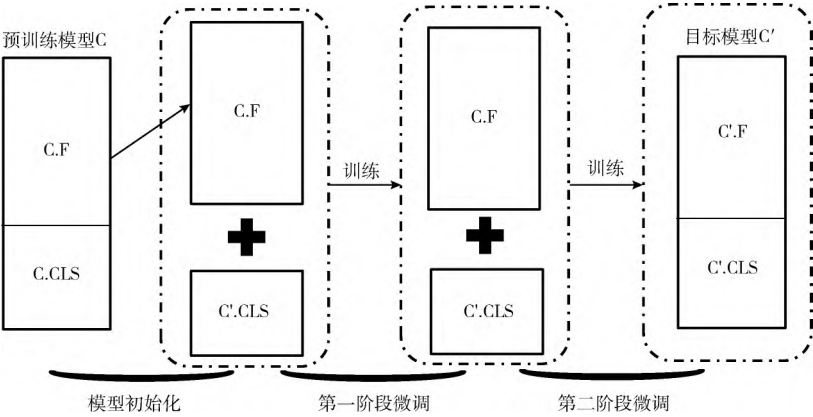


图 4 微调训练流程

3.3.2 消融实验

本节实验中研究了多种不同特征对分类网络性能的影响, 具体包括: Relu6 激活函数, 加入正则化项的损失函数, 标签平滑。实验中所用数据集为包含 22 500 张图片的数据集, 其中每个类别 2500 张图片, 共 9 个类别, 每个类别中 150 张图片被划分到测试集, 剩余图片用于网络训练。实验结果见表 1, 实验中基线模型使用 Relu 作为激活函数, 使用交叉熵损失函数进行训练, 未使用标签平滑技术。引入这 3 种特征后, 可以得到分类效果最好的模型。

3.4 风格迁移用于图像分类任务

风格迁移作为一种数据增强方法, 为了验证其在恶劣环境下对图像算法鲁棒性的影响, 本节使用风格迁移后的

图像扩充训练集, 并重新进行网络训练。实验中从每个类别随机选取 500 张图片进行风格迁移, 对转换后的图片进行一轮人工筛选, 共保留 350 张与自然图像较为接近的样本。

表 1 不同特征对分类网络准确率的影响

激活函数	正则化损失	标签平滑	准确率/%
			93.3
✓			93.3
✓	✓		93.5
✓	✓	✓	94

3.3.2 节中实验使用的测试集 test1 每个类别包含 150 张图片, 共 1350 张, 其中只包含少量恶劣环境样本。除此测试集外, 本节实验单独准备了只包含恶劣环境样本的测试集 test2, 这些样本均为恶劣环境下采集的自然图像, 未经过任何处理。由于恶劣环境下的样本采集难度较大, test2 中只包含 49 张测试图片。

表 2 中展示了使用风格迁移进行数据增强前后模型的分​​类效果。未使用风格迁移的训练集样本量为 21 150 张图片, 使用了风格迁移后, 在训练集中增加了数据增强样本 350 张。在 test1 数据集上, 数据增强前后模型的准确率相同, 但是在 test2 数据集上, 数据增强技术使得模型分类准确率提升了 8.2%, 说明风格迁移作为一种数据增强技术, 可以显著提升图像算法在恶劣环境下的鲁棒性。

表 2 风格迁移对模型鲁棒性的影响

风格迁移	训练集 样本量	数据集 1 测试 准确率/%	数据集 2 测试 准确率/%
	21 150	94	55.1
✓	21 150+350	94	63.3

上述实验结果中, 虽然风格迁移使模型在恶劣环境数据下的表现有了较大提升, 但是与正常环境下的分类效果相比, 准确率仍存在较大差距。为避免生成样本对模型识别正常图像的效果产生影响, 本实验中对于风格迁移算法生成的图像, 只将人工挑选出的与自然图像差异较小的样本用于训练, 相比于训练集中所有样本的数据规模, 生成样本的占比较小。增加恶劣环境的样本数量后, 模型在恶劣环境下的效果仍有较大提升空间。

4 结束语

本文使用图像风格迁移进行数据增强, 增加了难以采集的大雾、风沙条件下的样本量, 提升了模型在恶劣环境下的表现。在进行分类任务时, 使用 PeleeNet 作为网络骨架, 并对网络中用到的激活函数和损失函数进行了调整, 采用了标签平滑的训练策略, 提升了模型的分​​类效果。但是本文所用方法仍然存在风格迁移成功率较低、图像分类模型受恶劣环境影响较大等问题, 这些问题有待于在未来的研究中进一步解决。

参考文献:

- [1] Zhang Z. Improved Adam optimizer for deep neural networks [C] //IEEE/ACM 26th International Symposium on Quality of Service. Canada: IEEE, 2018: 1-2.
- [2] Ding J, Ren X, Luo R, et al. An adaptive and momental bound method for stochastic learning [J]. arXiv preprint: 2019, arXiv: 1910.12249.
- [3] Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. USA: IEEE, 2020: 1580-1589.
- [4] Zhang X, Zhou X, Lin M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. USA: IEEE, 2018: 6848-6856.
- [5] Tan M, Pang R, Le QV. Efficientdet: Scalable and efficient object detection [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. USA: IEEE, 2020: 10781-10790.
- [6] Tian Z, Shen C, Chen H, et al. Fcos: Fully convolutional one-stage object detection [C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Korea (South): IEEE, 2019: 9627-9636.
- [7] Ali A, Testa M, Bianchi T, et al. BioMetricNet: Deep unconstrained face verification through learning of metrics regularized onto Gaussian distributions [C] //European Conference on Computer Vision. Springer, Cham, 2020: 133-149.
- [8] Hwang DY, Hatzinakos D. PPG-based personalized verification system [C] //IEEE Canadian Conference of Electrical and Computer Engineering. Canada: IEEE, 2019: 1-4.
- [9] Wang RJ, Li X, Ling CX. Pelee: A real-time object detection system on mobile devices [C] //Advances in Neural Information Processing Systems. Montreal CANADA: MIT Press, 2018: 1963-1972.
- [10] Zhong Z, Zheng L, Kang G, et al. Random erasing data augmentation [C] //Proceedings of the AAAI Conference on Artificial Intelligence. USA: AAAI, 2020: 13001-13008.
- [11] Yang J. GridMask based data augmentation for Bengali handwritten grapheme classification [C] //Proceedings of the 2nd International Conference on Intelligent Medicine and Image Processing. China: ACM, 2020: 98-102.
- [12] Zhang H, Cisse M, Dauphin YN, et al. Mixup: Beyond empirical risk minimization [J]. arXiv preprint: 2017, arXiv: 1710.09412.
- [13] Li Y, Liu MY, Li X, et al. A closed-form solution to photo-realistic image stylization [C] //Proceedings of the European Conference on Computer Vision. Germany: Springer, 2018: 453-468.
- [14] Ghiasi G, Lee H, Kudlur M, et al. Exploring the structure of a real-time, arbitrary neural artistic stylization network [J]. arXiv preprint: 2017, arXiv: 1705.06830.
- [15] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint: 2014, arXiv: 1409.1556.
- [16] Howard AG, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications [J]. arXiv preprint: 2017, arXiv: 1704.04861.

- [17] Ioffe, Sergey, Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift [C] //Proceedings of the 32nd International Conference on Machine Learning. France: ACM, 2015: 448-456.
- [18] Liu Z, Li J, Shen Z, et al. Learning efficient convolutional networks through network slimming [C] //Proceedings of the IEEE International Conference on Computer Vision. Italy: IEEE, 2017: 2736-2744.
- [19] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. USA: IEEE, 2016: 2818-2826.
- [20] Müller R, Kornblith S, Hinton GE. When does label smoothing help? [C] //Advances in Neural Information Processing Systems. Vancouver CANADA: MIT Press, 2019: 4696-4705.
- [21] He T, Zhang Z, Zhang H, et al. Bag of tricks for image classification with convolutional neural networks [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. USA: IEEE, 2019: 558-567.
- [22] Kingma, Diederik P, Jimmy Ba. Adam: A method for stochastic optimization [J]. arXiv preprint; 2014, arXiv: 1412.6980.
- [23] He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification [C] //Proceedings of the IEEE International Conference on Computer Vision. Chile: IEEE, 2015: 1026-1034.