图像溯源算法简述

电信 1902 蔡鹏飞
201961204

一、数据预处理

首先将原数据每类按 8：1 的比例划分出训练集和验证集。注意到训练集图片大小远大于 512 * 512 的标准，因此我采用随机裁剪的方式从每一张原图片裁剪出 30 张 512*512 的子图片。为了进一步扩充数据集，并增强模型对噪声和压缩的鲁棒性，我从裁剪后的每一类随机取 300 张图片加高斯白噪声（噪声方差范围在 5-26 之间的均匀分布），又随机取 300 张图片以 85 的质量因子进行压缩。最终训练集每一类图片数为 3000，共 30，000 张训练样本，1800 张验证样本。

二、分类算法

本实验采用了 VGG16 进行分类，网络结构见图 1。最开始我尝试使用大小为 512 *512 的图片直接作为输入，但这样的问题在于输入为 512*512 大小的 vgg 网络参数实在太多了，为了使 GPU 能够容纳的下，每一批的 batch_size 需要设的很小，这提高了优化的难度。所以我采用的策略是将图片分为 4 张 252*252 的子图片进行训练，并将预测结果取平均作为最终结果。训练的轮数为 25 轮，优化方法为随机梯度下降法。

我同样也尝试了使用更浅的网络，如 AlexNet,以及在输入进行某些特殊预处理以增加先验的方式进行预测（如使用 LBP 预处理），但效果不佳。这从一定程度上反应了更深的卷积神经网络在图像溯源问题上具有更大的优势。

三、实验结果

验证集结果见图 2。其中在原始验证集、压缩验证集、加噪验证集的准确率分别为 0.987，0.984，0.835。可以看出模型在原始数据集和压缩数据集上取得了相当高的准确率，而在加噪数据集上表现不佳，可能因为最大方差为 26 的白噪声对模型来说确实难以识别，说明了模型对噪声较为敏感。

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 256, 256, 3)]	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0
global_average_pooling2d_1 (	(None, 512)	0
flatten_1 (Flatten)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 10)	2570

图 1 分类所用的网络结构

精确度0.9871345029239766
0-Apple_iPhone6Plus：accuracy=0.9351
1-Canon_PowerShotA640：accuracy=0.9878
2-Sony_DSC-W170：accuracy=1.0
3-Samsung_GalaxyS5：accuracy=0.9778
4-Huawei_P9：accuracy=1.0
5-Nikon_D70s：accuracy=1.0
6-OnePlus_A3003：accuracy=0.9888
7-Microsoft_Lumia640LTE：accuracy=1.0
8-Lenovo_P70A：accuracy=0.989
9-Xiaomi_RedmiNote3：accuracy=0.9889

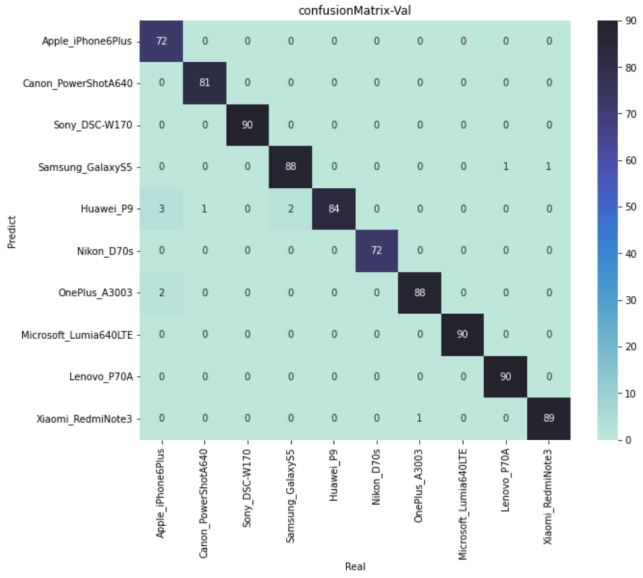

图 2.1 在原始图片验证集上的测试结果

精确度0.9836257309941521
0-Apple_iPhone6Plus：accuracy=0.9595
1-Canon_PowerShotA640：accuracy=0.9873
2-Sony_DSC-W170：accuracy=0.9783
3-Samsung_GalaxyS5：accuracy=0.9674
4-Huawei_P9：accuracy=1.0
5-Nikon_D70s：accuracy=0.973
6-OnePlus_A3003：accuracy=0.9888
7-Microsoft_Lumia640LTE：accuracy=1.0
8-Lenovo_P70A：accuracy=0.989
9-Xiaomi_RedmiNote3：accuracy=0.9889

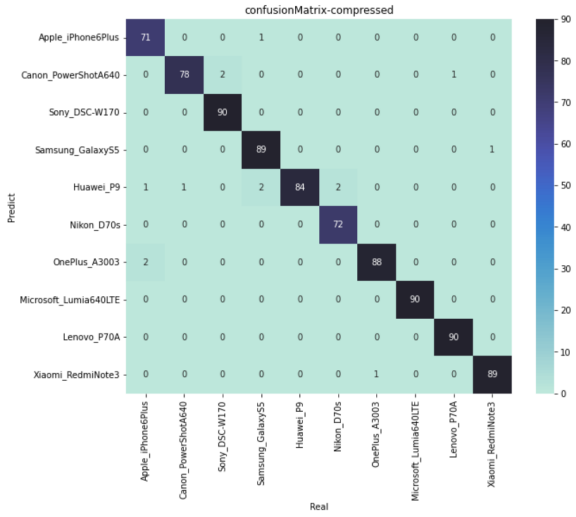

图 3.2 在压缩图片验证集上的测试结果

精确度0.8350877192982457
0-Apple_iPhone6Plus：accuracy=0.7534
1-Canon_PowerShotA640：accuracy=0.974
2-Sony_DSC-W170：accuracy=0.8866
3-Samsung_GalaxyS5：accuracy=0.7364
4-Huawei_P9：accuracy=0.9028
5-Nikon_D70s：accuracy=0.8333
6-OnePlus_A3003：accuracy=0.8108
7-Microsoft_Lumia640LTE：accuracy=0.8068
8-Lenovo_P70A：accuracy=0.8235
9-Xiaomi_RedmiNote3：accuracy=0.8556

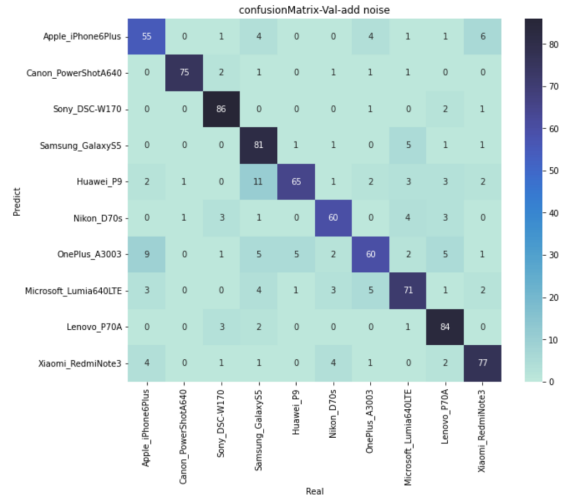

图 4.3 在加噪验证集上的测试结果

图 2 验证集结果