|  |  |  |
| --- | --- | --- |
| 第四章 编程框架实践 | | |
| 姓名： 吴文韬 | 学号：20373914 | 班级： 212111 |
| **注意：课程实验代码和实验报告需独立完成，严禁抄袭!** 实验4.1基于VGG19实现图像分类(40’’) **1.本实验与实验二中使用Python实现的图像分类相比，在识别精度、识别速度上有哪些差异？**(20’’)  **纯CPU**  **CleanShot 2023-11-27 at 21.01.00**  **使用TensorFlow框架**  **CleanShot 2023-11-27 at 21.03.02**  **精度方面略有提高，但是识别速度要快10倍不止**  **2.为什么会有这些差异？**(20’’)   1. TensorFlow框架硬件加速：框架内置了很多优化算法和硬件加速功能，例如自动微分、GPU加速等。这些功能可以显著提高模型的训练和预测速度 2. TensorFlow框架提供了预训练的VGG19模型，这个模型在大量数据上进行了训练，因此在识别精度上可能会比直接使用Python实现的VGG19更高  实验4.2 实时图像风格迁移(60’’) **1. 简要介绍实例归一化（IN）和批归一化（BN），以及两者的区别？**(20’’)  IN：每个样本上进行归一化，即对每个样本的每一层，按照其本身的均值和方差进行归一化  BN：在一个mini-batch的数据上进行归一化，即对每一层的mini-batch数据，按照其本身的均值和方差进行归一化  区别：在于归一化的范围不同，因此BN在处理小批量数据时会出现性能下降的情况，而IN则不会  **2. 在调用TensorFlow内置的卷积以及转置卷积函数tf.nn.conv2d()、tf.nn.conv2d\_transpose()时，边缘扩充方式选择“SAME”和“VALID”有什么区别？对生成图像结果有何影响？**(20’’)  **"SAME"会在输入的边缘添加足够的零，使得卷积后的输出和输入有相同的尺寸，但可能会引入一下噪声**  **"VALID"不会添加任何额外的零，只对输入的有效部分进行卷积，这种方式会使卷积后的图像尺寸小于卷积前的，但不会引入噪声**  **3. 本实验中实时风格迁移和实验二中的非实时风格迁移在算法设计和运行效率上有什么不同？**(20’’)  算法设计：  非实时风格迁移：使用梯度下降来最小化内容损失和风格损失。这需要迭代多次，每次迭代都需要计算损失并更新图像。这种方法可以得到高质量的结果，但计算成本高，不能实时运行。  实时风格迁移：使用预训练的神经网络来进行风格迁移。这种方法只需要一次前向传播，可以实时运行，但可能无法达到非实时风格迁移的质量。  运行效率：  非实时风格迁移：由于需要迭代多次，因此运行效率较低，不能实时运行。  实时风格迁移：由于只需要一次前向传播，因此运行效率高，可以实时运行。 | | |