**深 圳 大 学 实 验 报 告**

**课程名称： 人工智能课程实训**

**实验项目名称： 实验3模型优化实践**

**学院： 计算机与软件学院**

**专业： 软件工程（腾班）**

**指导教师： 王旭**

**报告人： 黄亮铭 学号： 2022155028 班级： 腾班**

**实验时间： 2024年10月17日至11月06日**

**实验报告提交时间： 2024年11月 06日**

**教务处制**

|  |
| --- |
| **实验目的与要求：**  **目标：**   1. 了解模型训练前数据预处理和增强方法； 2. 掌握对象检测模型YOLO的训练与优化基本方法； 3. 掌握有方向的YOLO-OBB 模型的基本原理与优化方法。   **基本要求：**   1. 将众包完成的口算数据集通过脚本/代码处理的方式，完成数据集的准备，训练集/测试集的比例划分； 2. 在训练前完成预处理操作，不仅限于翻转，加噪等视觉数据增强方法； 3. 基于YOLO预训练模型在口算数据集进行模型微调训练，在实验内容中给出预处理前后不同模型在测试集上的性能对比。 4. 使用Weights & Biases工具完成模型训练的可视化。   **高级要求：**   1. 通过代码处理的方式对数据集中的图片进行旋转，扭曲等几何失真（opencv库中图像处理仿射变换等算子）仿真，以模仿真实场景的拍摄失真。 2. 基于YOLO-OBB模型在增强后的数据集上进行模型微调训练，在实验内容中给出预处理前后不同模型在测试集上的性能对比。 |
| **方法、步骤：**  数据集的准备 Xml格式转txt格式  * 实现思路：因为YOLOv5模型对输入数据的要求，所以我们需要将xml格式的文件转化为txt格式的文件。YOLOv5的xml文件的格式如下：。YOLOv5的txt文件的格式如下：。由此可以得出从xml文件转换为txt文件的坐标计算公式：   计算完上述值后还需要归一化，即除以图像的宽度和高度。  对于，因为我们需要识别的均为口算题，所以只有一类，的值为0。  综上所述，我们需要在xml格式的文件中找到对应的标签的值然后根据公式转化为txt格式的文件即可。 划分训练集和测试集  1. 实现思路：首先获取所有图像和标签文件的列表，然后遍历图像文件列表，按照8：2的比例将文件随机划分到训练集或测试集中，同时根据图像文件的文件名找到对应的标签文件，将标签文件也放到相应的集合中。  |  | | --- | |  | | 划分训练集和验证集  1. 实现思路：首先获取训练集的图像和标签文件的列表，然后遍历图像文件列表，按照3：1的比例将文件随机划分到新的训练集或验证集中，同时根据图像文件的文件名找到对应的标签文件，将标签文件也放到相应的集合中。 |   预处理——数据增强   1. 实现思路：首先利用OpenCV读入所有照片。对于每一张图片，25%几率增加高斯噪声，25%几率增加椒盐噪声，50%几率不增加噪声。同时，对于每一张图片，分别有25%几率水平和垂直翻转，50%几率不进行任何翻转操作。最后，将照片输出到指定目录。   模型微调训练   1. 我们需要训练两个模型：一个在原始口算题数据集上进行训练，另一个在数据增强后的口算题数据集上进行训练。 2. 在进行模型微调训练之前，我们需要修改如下配置文件。  * 文件：1）修改数据集的根路径；2）修改训练集的路径（相对路径）；3）修改验证集的路径（相对路径）；4）添加：；5）修改类别文件：因为我们只有算式一种需要识别的物体，所以类别只有一种就是。 * 文件：修改为。  1. 在train.py文件中修改训练参数。  * 模型选择参数：指定。 * 配置文件参数：指定。 * 数据集参数：指定。 * 训练轮数：将默认值设置为100。 * Device参数：选择0，使用显卡进行训练。  1. 在命令行中输入命令开始训练。训练结束后会在相应的文件目录下生成最佳的模型参数和最后一次的模型参数。此外还会有示例图片以及部分参数可视化。 2. 训练结果展示。   模型训练可视化   1. 根据实验文档，我们选择Weights & Biases工具进行模型训练可视化。 2. 首先在命令行中输入命令：安装模型训练可视化的包。 3. 然后输入命令：登录Weights & Biases，接下来输入自己的API KEY。 4. 最后在train.py文件中对模型可视化训练项目进行命名。 5. 以原始数据训练过程可视化为例进行展示。   模型测试与性能对比 预处理前模型测试  1. 实现思路：首先YOLOv5中有几种已经预训练好的模型，我选择了YOLOv5s模型。然后进入detect.py文件修改参数。  * 模型选择参数：指定在原始口算题训练集上训练的模型参数。 * 数据源参数：指定测试集目录。  预处理后模型测试  1. 实现思路：首先YOLOv5中有几种已经预训练好的模型，我选择了YOLOv5s模型。然后进入detect.py文件修改参数。  * 模型选择参数：指定在数据增强口算题训练集上训练的模型参数。 * 数据源参数：指定测试集目录。   模仿真实场景的拍摄失真  实现思路：   1. 首先利用读入所有图像。 2. 然后对于每一张图片，1/3的概率进行旋转处理（在-180°到180°随机旋转），1/3概率进行扭曲处理，1/3概率不进行任何处理。 3. 最后，将图片输出到对应的目录。   基于YOLO-OBB模型微调训练  因为YOLO-OBB模型是基于YOLO模型发展的，所以他们之间的训练参数的调整方法也高度相似。因此，我这里直接给出模型在数据增强前后的数据集训练后的测试结果以及它们之间的性能对比。 |
| **实验过程及内容：**  数据集的准备 Xml格式转txt格式  1. 实现思路：因为YOLOv5模型对输入数据的要求，所以我们需要将xml格式的文件转化为txt格式的文件。YOLOv5的xml文件的格式如下：。YOLOv5的txt文件的格式如下：。由此可以得出从xml文件转换为txt文件的坐标计算公式：   计算完上述值后还需要归一化，即除以图像的宽度和高度。  对于，因为我们需要识别的均为口算题，所以只有一类，的值为0。  综上所述，我们需要在xml格式的文件中找到对应的标签的值然后根据公式转化为txt格式的文件即可。   1. 核心代码：实现这一步骤的核心代码如下图所示。  |  | | --- | |  | | 图1a：根据公式计算的函数 | |  | | 图1b：根据标签找到值的函数 |  划分训练集和测试集  1. 实现思路：首先获取所有图像和标签文件的列表，然后遍历图像文件列表，按照8：2的比例将文件随机划分到训练集或测试集中，同时根据图像文件的文件名找到对应的标签文件，将标签文件也放到相应的集合中。 2. 核心代码：实现这一步骤的核心代码如下图所示。  |  | | --- | |  | | 图2：划分训练集和测试集的核心代码 划分训练集和验证集  1. 实现思路：首先获取训练集的图像和标签文件的列表，然后遍历图像文件列表，按照3：1的比例将文件随机划分到新的训练集或验证集中，同时根据图像文件的文件名找到对应的标签文件，将标签文件也放到相应的集合中。 2. 核心代码：实现这一步骤的核心代码如下图所示（与上面划分训练集和测试集的代码相同）。     图3：划分训练集和测试集的核心代码 |   预处理——数据增强   1. 实现思路：首先利用OpenCV读入所有照片。对于每一张图片，25%几率增加高斯噪声，25%几率增加椒盐噪声，50%几率不增加噪声。同时，对于每一张图片，分别有25%几率水平和垂直翻转，50%几率不进行任何翻转操作。最后，将照片输出到指定目录。 2. 核心代码：实现这一步骤的核心代码如下图所示。  |  | | --- | |  | | 图4a：高斯噪声函数 | |  | | 图4b：椒盐噪声函数 | |  | | 图4c：；图像翻转函数   1. 添加噪声和翻转图像前后对比。  |  |  | | --- | --- | |  |  |   图4d：对比（左：数据增强前 右：数据增强后） |   模型微调训练   1. 我们需要训练两个模型：一个在原始口算题数据集上进行训练，另一个在数据增强后的口算题数据集上进行训练。 2. 在进行模型微调训练之前，我们需要修改如下配置文件。  * 文件：1）修改数据集的根路径；2）修改训练集的路径（相对路径）；3）修改验证集的路径（相对路径）；4）添加：；5）修改类别文件：因为我们只有算式一种需要识别的物体，所以类别只有一种就是。     图5：文件（修改后）   * 文件：修改为。     图6：文件（修改后）   1. 在train.py文件中修改训练参数。  * 模型选择参数：指定。 * 配置文件参数：指定。 * 数据集参数：指定。 * 训练轮数：将默认值设置为100。 * Device参数：选择0，使用显卡进行训练。       图7：修改训练参数   1. 在命令行中输入命令开始训练。训练结束后会在相应的文件目录下生成最佳的模型参数和最后一次的模型参数。此外还会有示例图片以及部分参数可视化。     图8：训练过程   1. 训练结果展示。  |  |  | | --- | --- | |  |  |   图9：训练结果（左：原始数据集 右：数据增强数据集）  模型训练可视化   1. 根据实验文档，我们选择Weights & Biases工具进行模型训练可视化。 2. 首先在命令行中输入命令：安装模型训练可视化的包。     图10：安装wandb   1. 然后输入命令：登录Weights & Biases，接下来输入自己的API KEY。     图11：登录wandb   1. 最后在train.py文件中对模型可视化训练项目进行命名。     图12：项目命名   1. 以原始数据训练过程可视化为例进行展示。  |  |  | | --- | --- | |  | | | 图13a：衡量标准 | | |  | | | 图13b：训练损失和验证损失 | | |  | | | 图13c：GPU变化情况（部分） | | |  | | | 图13d：P\_curve | | |  | | | 图13e：PR\_curve | | |  | | | 图13f：R\_curve | | |  |  | | 图13g：label | |   模型测试与性能对比 预处理前模型测试  1. 实现思路：首先YOLOv5中有几种已经预训练好的模型，我选择了YOLOv5s模型。然后进入detect.py文件修改参数。  * 模型选择参数：指定在原始口算题训练集上训练的模型参数。 * 数据源参数：指定测试集目录。  1. 参数调整：调整的参数为最佳模型参数保存的目录，如下图所示。     图14：修改模型参数目录   1. 在命令行中运行命令：开始测试。     图15：测试完成   1. 部分结果展示。可以看到，在原始数据上进行训练的模型在遇到正常情况时的表现比较好，识别率较高；但是在遇到椒盐噪声和高斯噪声等噪声影响的情况和图像翻转的情况时模型的表现较差，识别率急剧下降。  |  |  | | --- | --- | |  |  | | 图16a：正常 | 图16b：翻转 | |  |  | | 图16c：噪声（重度） | 图16d：噪声（轻度） |  预处理后模型测试  1. 实现思路：首先YOLOv5中有几种已经预训练好的模型，我选择了YOLOv5s模型。然后进入detect.py文件修改参数。  * 模型选择参数：指定在数据增强口算题训练集上训练的模型参数。 * 数据源参数：指定测试集目录。  1. 参数调整：调整的参数如下图所示。     图17：修改模型参数目录   1. 在命令行中运行命令：开始测试。     图18：测试完成   1. 部分结果展示。可以看到，在数据增强后的数据集上训练的模型具有较强的抗干扰能力。在正常情况、翻转情况和轻度噪声情况等三种情况下的表现均比较好，识别率较高（图19a、b、d）。即使是在重度噪声的情况，仍然具有一定的识别能力。  |  |  | | --- | --- | |  |  | | 图19a：正常 | 图19b：翻转 | |  |  | | 图19c：噪声（重度） | 图19d：噪声（轻度） |  性能对比 预处理前的模型的部分测试结果如图16所示，预处理后的模型的部分测试结果如图17所示。由两部分图（共8张图）进行对比发现，在正常情况下，两个模型对口算题的正确识别率相差不大，均为较高值。但是在有翻转和噪声的情况下，预处理前的模型会出现大量误识别和错识别的情况，口算题的正确识别率急剧下降；而预处理后的模型抗噪声能力会比较强，虽然有部分误识别和错识别的情况，但整体正确识别率仍然较高。特别是在重度噪声的情况，预处理前后的模型的性能差距非常大。  由上分析可知，预处理后的模型性能优于预处理前的模型。  模仿真实场景的拍摄失真  实现思路：   1. 首先利用读入所有图像。 2. 然后对于每一张图片，1/3的概率进行旋转处理（在-180°到180°随机旋转），1/3概率进行扭曲处理，1/3概率不进行任何处理。 3. 最后，将图片输出到对应的目录。   核心代码：   1. 决定是旋转图片还是扭曲图片，亦或是不做处理。     图20：选择分支   1. 旋转图片并保存的函数。     图21：旋转图片函数   1. 扭曲图片并保存的函数。     图22：扭曲图片函数  处理前后图片对比：   |  |  | | --- | --- | |  |  | | a:正常 | b:旋转 | |  |  | | c:正常 | d:扭曲 |   图23：正常图片与旋转、扭曲的图片对比  基于YOLO-OBB模型微调训练  因为YOLO-OBB模型是基于YOLO模型发展的，所以他们之间的训练参数的调整方法也高度相似。因此，我这里直接给出模型在数据增强前后的数据集训练后的测试结果以及它们之间的性能对比。   |  |  | | --- | --- | |  |  |   图24：预处理前后对比（左：预处理前 右：预处理后）  可以看到在预处理前，模型识别率非常低；在预处理之后，模型的识别率大大增大，虽然错识别率也上升了。 |
| **实验结论：**   * 本次实验我完成了基本要求的内容，即划分数据集、数据增强（预处理）、模型微调训练和训练过程可视化。 * 本次实验我完成了高级要求中的模仿真实场景的拍摄失真。 * 对于高级要求的基于YOLO-OBB模型微调训练，因为模仿真实场景的拍摄失真即扭曲和旋转后标签和口算题的实际位置不符，所以训练效果不尽人意。个人能力有限，无法对数据标签进行修改。 |
| **心得体会：**   * 通过本次实验，我了解了如何对数据集进行划分。 * 通过本次实验，我了解模型训练前数据预处理和增强方法，使得训练的模型更加健壮。 * 通过本次实验，我基本掌握了对象检测模型YOLO的训练方法。 * 通过本次实验，我基本掌握了YOLO-OBB 模型的基本原理与使用方法。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：王旭  2024 年 11 月07 日 |
| 备注： |