

第 3 章

边缘计算与人工智能模型开发

本章结合 AiCam 平台学习边缘计算与人工智能模型开发全流程。本章内容包括：

(1) 数据采集与标注：掌握数据采集和标注的过程，以及基于 CreateImg 进行图像采集、基于 labelImg 进行图像标注的过程。

(2) YOLOv3 模型的训练与验证：掌握模型训练和验证的作用，了解主流深度学习开发框架、深度学习目标检测算法原理、YOLO 的原理，结合 YOLOv3 和 Darknet 框架对交通标志识别模型进行训练与验证。

(3) YOLOv5 模型的训练与验证：掌握 YOLOv5 原理，结合 PyTorch 框架和 YOLOv5 对口罩检测模型进行训练与验证。

(4) YOLOv3 模型的推理与验证：掌握从开发框架到推理框架的转换过程，了解典型的常用移动端边缘推理框架，结合 YOLOv3 和 NCNN 框架对交通标志识别模型进行推理与验证。

(5) YOLOv5 模型的推理与验证：掌握 RKNN 框架的工作原理，结合 RKNN 框架和 YOLOv5 对口罩检测模型进行推理与验证。

(6) YOLOv3 模型的接口应用：了解常用的模型接口，掌握 NCNN 框架的模型接口设计，结合 YOLOv3 和 NCNN 框架设计交通标志识别模型的接口。

(7) YOLOv5 模型的接口应用：掌握 RKNN 框架的模型接口设计，结合 YOLOv5 和 RKNN 框架设计口罩检测模型的接口。

(8) YOLOv3 模型的算法设计：掌握 AiCam 平台的开发框架，结合 AiCam 平台和 YOLOv3 设计交通标志识别算法。

(9) YOLOv5 模型的算法设计：结合 AiCam 平台和 YOLOv5 设计交通标志识别算法。

3.1 数据采集与标注

数据采集与标注是深度学习和人工智能开发中的关键步骤，对构建高性能、鲁棒性强的模型至关重要。数据采集与标注的作用如下：

(1) 训练模型：深度学习需要大量的数据来训练模型，以便从中学到模式和规律。通过采集并标注有代表性的数据集，可以帮助模型更好地理解输入数据的特征和关系。

(2) 验证模型：采集的数据集可以用来验证模型的性能，标注的数据集可以用来测试模型的准确性、鲁棒性和泛化能力，从而评估模型在真实世界中的表现。

(3) 改进模型：通过分析模型在采集和标注的数据集上的表现，可以识别模型的弱点和

错误，并进一步优化和改进模型的性能，有助于不断提升模型的质量。

(4) 解决样本偏差：数据采集与标注有助于解决样本偏差问题，通过在数据集中包含各种各样的样本，可以更好地确保模型在各种情境下都能有良好的表现。

(5) 应对标签不平衡：在某些应用中，不同类别的样本可能具有不平衡的标签分布，通过采集并标注标签分布平衡的数据，可以帮助模型更好地应对标签的不平衡。

本节的知识点如下：

- 掌握数据采集与标注的过程。
- 结合边缘计算平台掌握通过 CreateImg 采集图像的步骤。
- 结合边缘计算平台掌握通过 labelImg 标注图像的步骤。

3.1.1 原理分析与开发设计

3.1.1.1 数据采集和标注流程

数据采集和标注需要确保数据的质量和准确性，从而为边缘视觉应用提供可靠的数据基础。数据采集和标注的一般流程如下：

(1) 确定数据需求：在进行数据采集和标注之前，要明确所需数据的类型和数量，并确保数据和业务需求、算法模型要求相匹配。

(2) 收集原始数据：根据需求收集原始数据，原始数据可以来自各种渠道，如互联网、传感器、摄像头等。

(3) 清洗数据：对原始数据进行清洗，去除噪声、重复和无效数据，确保数据的质量和准确性。

(4) 标注数据：将清洗后的数据标注为所需的类别或属性，以便算法模型能够识别和学习。标注可以手动进行或自动进行。手动标注需要专业的标注人员，自动标注需要训练好的算法模型。

(5) 验证数据：数据标注完成后，需要对数据进行验证和校对，以确保数据的一致性和准确性。

(6) 存储和管理数据：数据验证完成后，需要将数据存储于数据库或云端，以便进行管理。

(7) 更新和迭代数据：数据采集与标注是一个持续进行的过程，随着业务需求和算法模型的变化，需要不断更新和迭代数据，以保证数据的时效性和准确性。

3.1.1.2 图像采集

CreateImg 是一款可以自动采集图像工具，该工具通过 OpenCV 调用摄像头自动进行视频的录制和图像的采集，数据采集完成后被保存在 AllLabelImg/dataset/JPEGImages 文件夹内。

录制的视频要尽量在真实场景中进行，将需要检测的物品放置在摄像头能够拍摄到的视场位置，并在不同方位、不同角度、不同光线的条件下录制物品，尽量保持数据样本的多样性。

3.1.1.3 图像标注

labelImg 是一个可视化的图像标注工具，Faster R-CNN、YOLO、SSD 等目标检测网络所需的数据集，均可采用 labelImg 标注的图像。labelImg 在完成数据标注后会自动生成描述图像的 XML 文件，生成的 XML 文件遵循 PASCAL VOC 格式标准。

3.1.2 开发步骤与验证

3.1.2.1 图像采集

将摄像头接入边缘计算或者计算机，通过 CreateImg 采集图像的步骤如下：

(1) 将需要识别的物品放置在摄像头正前方的合适位置。

(2) 打开终端，输入以下命令运行 CreateImg：

```
#Linux 环境下执行以下命令
$ conda activate py36_tf114_torch15_cpu_cv345           //Ubuntu 20.04 操作系统下需要切换环境
$ python3 CreateImg.py

#Windows 环境下执行以下命令（需要安装必要的环境）
$ python CreateImg.py
```

(3) CreateImg 将显示摄像头的实时图像（见图 3.1），并将实时图像保存到数据集目录 AllLabelImg\dataset\JPEGImages。



图 3.1 摄像头的实时图像

(4) 在不同方位、不同角度、不同光线下拍摄物品，尽量使数据样本保持多样性。

(5) 录制一段时间后，按下“Q”退出 CreateImg，在数据集目录中可以看到采集的图像。

3.1.2.2 图像标注

1) 导入图像

(1) 进入 labelImg 所在的目录，运行 labelImg.exe。labelImg 的运行界面如图 3.2 所示。

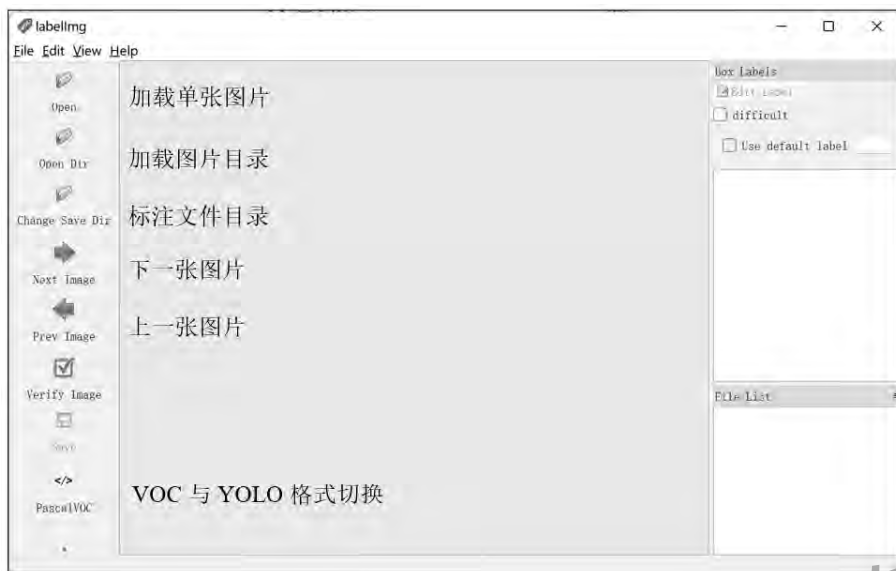


图 3.2 labelImg 的运行界面

(2) 单击“Open Dir”按钮，选择采集的图像目录 AILabelImg\dataset\JPEGImages（可以使用前面采集的数据集图像，也可以使用本项目代码内提供的交通标志数据集图像），如图 3.3 所示。

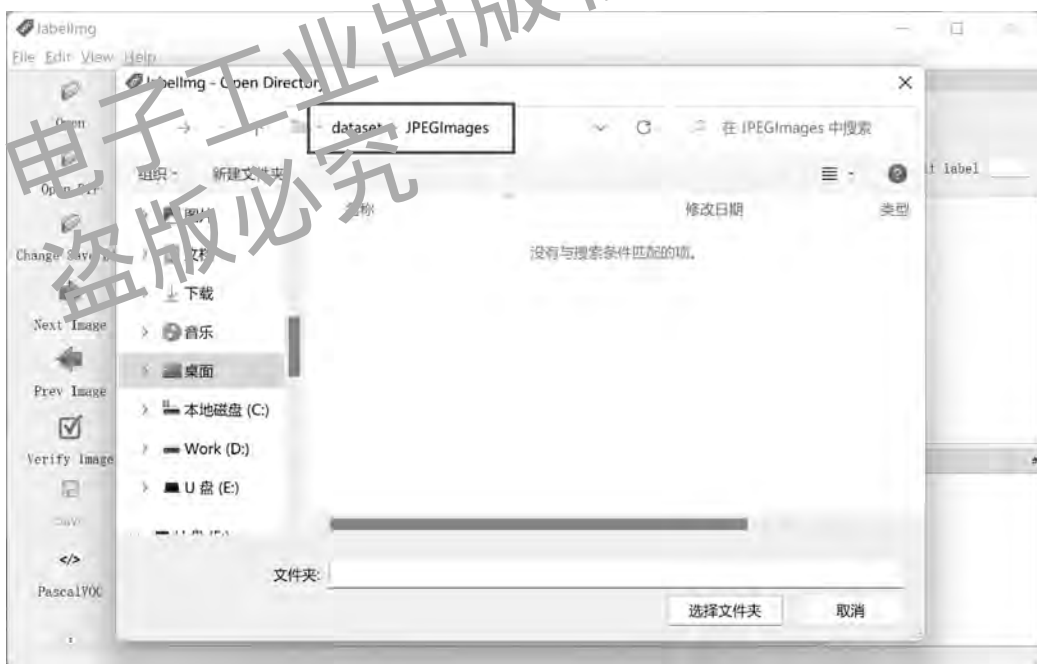


图 3.3 选择采集的图像目录

(3) labelImg 会自动弹窗，要求选择标注文件的存储路径 AILabelImg\dataset\Annotations，如图 3.4 所示。



图 3.4 选择标注文件的存储路径

(4) 设置完成后, labelImg 将加载图像并在文件列表 (File List) 中显示加载的图像, 单击第一幅图像将会在 labelImg 运行界面的中间显示该图像。加载图像如图 3.5 所示。



图 3.5 加载图像

(5) 在 labelImg 运行界面中选择菜单 “View” → “Auto Saving”, 如图 3.6 所示, 即可设置为自动保存标注文件。



图 3.6 选择菜单“View”→“Auto Saving”

2) 图像标注

(1) 图像标注: 在 labelImg 运行界面的文件列表中单击第一幅图像, labelImg 将开始逐幅对图像进行标注。按“W”键可对目标物品进行矩形框标注, 在需要标注的目标左上角按下鼠标左键并且不要松开, 移动到目标的右下角后松开鼠标左键, 即可对一个目标进行标注, 如图 3.7 所示。



图 3.7 图像标注

(2) 设置标签 (Label): 勾选“Box Labels”中的“Edit Label”, 在弹出的“labelImg”对话框输入标签名称, 如 right (再画矩形框时就会有记录供直接选择), 单击“OK”按钮即可完成图像标注, 如图 3.8 所示。



图 3.8 设置标签后完成图像标注

(3) 默认标签: 勾选“Box Labels”中的“Use default label”, 可以对同一个标签进行批量标注, 这样在每次画矩形框后就会自动将标注设置为默认的标签, 从而避免每次添加标签, 如图 3.9 所示。



图 3.9 设置默认的标志

(4) 连续标注：按“D”键切换到下一幅图像，继续按“W”键可进行图像标注，如图 3.10 所示。



图 3.10 连续标注

(5) 标注技巧：读者可以首先将其中一个标签设置为默认的标志，把所有图像内的该标签全部标注一次；然后将其他的标签设置为默认的标志，继续从头开始将所有图像中新的标签全部标注一次，直到图像中所有的标签都进行了标注，如图 3.11 所示。