

TensorFlow + OpenVINO™ 工具套件

一、概述

TensorFlow 是 Google 开源深度学习框架，可以用于机器学习、语音识别、目标检测等多种人工智能算法的开发。TensorFlow Object Detection API 是 TensorFlow 框架中专门用于目标检测应用的深度学习框架，使用该框架可以快速训练出不同种类的深度学习目标检测模型。如图 1 所示。

TensorFlow Object Detection API

TensorFlow 2.2 TensorFlow 1.15 Python 3.6

Creating accurate machine learning models capable of localizing and identifying multiple objects in a single image remains a core challenge in computer vision. The TensorFlow Object Detection API is an open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models. At Google we've certainly found this codebase to be useful for our computer vision needs, and we hope that you will as well.



图 1 TensorFlow Object Detection API

本文基于 TensorFlow Object Detection API 框架搭建基于深度学习的目标检测应用的开发环境，OpenVINO™工具套件为目标检测模型优化部署的框架，介绍了 TensorFlow Object Detection API 框架和 OpenVINO™工具套件的安装和使用，以及 FasterRCNN 目标检测模型训练的全部流程。并详细介绍了 OpenVINO™ Inference Engine 应用程序典型开发流程，以及怎样使用 C++ 编程语言开发 AI 推理应用程序。

训练目标检测模型需要极大的算力，本文的硬件平台选择为英特尔®NUC(Next Unit of Computer) 产品线系列的幻影峡谷，英特尔®NUC 是英特尔®公司设计制造的功能强大的迷你计算机(Mini PC)，而该系列中的幻影峡谷是一款具备 AI 训练能力的迷你计算机。

使用的幻影峡谷与深度学习模型训练相关的参数如下。

- 11th Gen Intel® Core™ i7-1165G7
- Intel® Iris® Xe Graphics
- NVIDIA GeForce RTX 2060

本文需要的软件运行环境及其对应版本如下。

- TensorFlow 2.4.0
- OpenVINO 2021.4 LTS
- Python 3.8.10

二、使用 TensorFlow Object Detection API 进行模型训练

第一步，标注数据集并生成 TFRecord 文件

首先从 Kaggle 直接下载猫狗数据集，使用 LabelImg 进行标注，讲标注后的 .xml 文件和 .jpg 格式的原文件一齐放进数据训练集所对应的文件夹，再按照比例依次划分验证集和测试机的比例并分别放入相应文件夹。

标注图片完毕后，根据标注文件的标签类型创建标签映射文件：label_map.pbtxt，在标签文件中，每个标注类型对应一个 item 项，每个项有两个属性，id 和 name。id 是整数数据本文标签有猫和狗两个类型，id 的编号便从 1 到 2，如代码清

单 1 所示:

代码清单 1 建标签映射文件

```
item {
  id: 1
  name: 'cat'
}

item {
  id: 2
  name: 'dog'
}
```

最后创建 TFRecord 文件, TFRecord 格式文件是向 TensorFlow 模型输入数据速度最快,效率最高的格式文件。标注完数据集,创建标签映射文件后,根据如图 2 所示相应命令,创建 train.tfrecord 和 eval.tfrecord 文件。

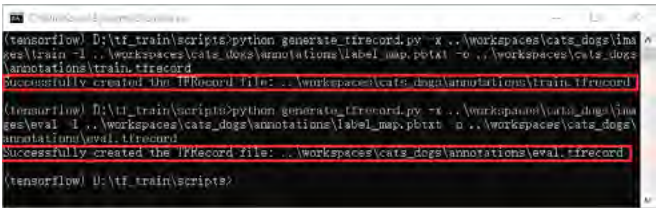


图 2 创建 TFRecord 文件

第二步, 修改预训练模型配置文件, 启动模型训练

TensorFlow Model Zoo 提供多个在 coco 数据集上完成训练的目标检测模型, 本文以下载 faster_rcnn_resnet50_v1_640*640_coco17_tpu-8 模型为例, 模型下载解压后, 得到“checkpoint”、“saved_model”和“pipeline.config”三个文件, 在进行迁移学习配置模型文件是需要修改“pipeline.config”文件的内部参数。

根据模型训练的需要分别修改配置文件“pipeline.config”中 num_classes、batch_size、num_steps 等参数值, 以及配置标签映射文件和 TFRecord 文件的路径。Pipeline.config 文件配置完毕后, 即可根据本文提供的模型训练脚本启动训练, 如图 3 所示。

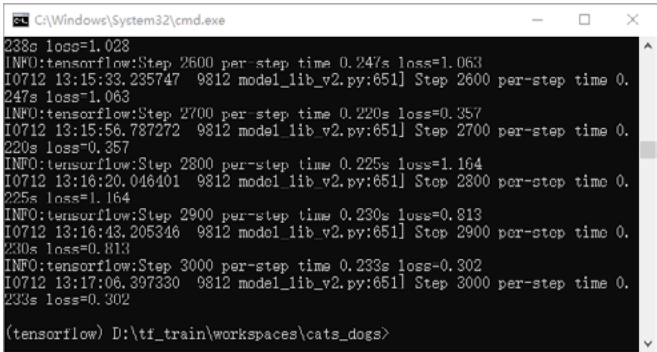


图 3 启动模型训练

第三步, 导出模型推理预测

在模型训练过程中会产生一系列的 checkpoint 检查点文件, 模型训练过程中, 所涉及到的参数和权重值全部记录在检查点文件中, 但是检查点文件没有关于模型计算的源代码。所以在模型推理计算阶段, 需要将 checkpoint 文件和模型计算的描述一起打包成可以部署的 SavedModel 格式。生成的 SavedModel 格式文件夹如图 4 所示。

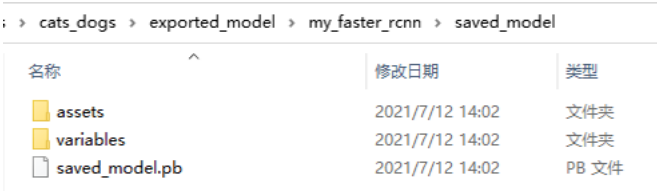


图 4 生成 SavedModel 格式文件

生成 SavedModel 后即可执行推理计算, 首先将本文附带的 object_detection_example_2.py 模型推理计算的脚本下载到本地, 打开 object_detection_example_2.py 文件, 根据实际路径情况修改 PATH_TO_IMAGES_DIR,PATH_TO_SAVED_MODEL,PATH_TO_LABELS 三处, 修改完毕运行该 Python 脚本, 即可得到模型推理计算的结果, 如图 5 所示。

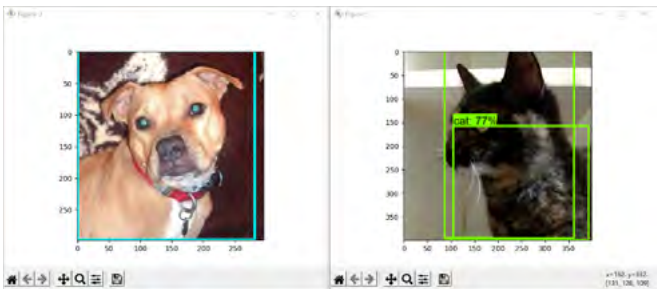


图 5 执行模型推理计算

三、使用 OpenVINO™ 工具套件部署

TensorFlow 支持将训练好的深度学习模型通过 OpenVINO™ 工具套件对模型进行优化部署，在初始化 OpenVINO™ 工具套件使用环境，安装完毕 OpenVINO™ 工具套件相关依赖后即可进行加速部署。

第一步，使用 Model—Optimizer 优化模型

Model_Optimizer 工具是 OpenVINO™ 工具套件中用于深度学习优化的 Python 脚本，支持从各种深度学习框架导入训练后的模型，并将其转换为 IR 格式文件。

启动 tensorflow 虚拟环境，首先输入命令 <pip install -r requirements_tf2.txt> 安装 mo_tf.py 运行时依赖的 python 软件包，再输入命令：<python mo_tf.py --saved_model_dir d:\tf_train\workspaces\cats_dogs\exported_model\my_faster_rcnn\saved_model --tensorflow_object_detection_api_pipeline_config d:\tf_train\workspaces\cats_dogs\exported_model\my_faster_rcnn\pipeline.config --output_dir d:\tf_train\workspaces\cats_dogs\IR_model --data_type FP16 --transformations_config extensions\front\tf\faster_rcnn_support_api_v2.0.json>，执行 TensorFlow 模型并优化成 IR 文件，运行结果如图 6 所示：

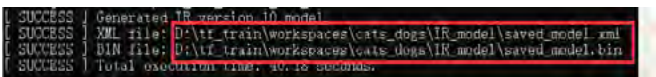


图 6 生成 IR 格式文件

第二步，Inference Engine 应用程序典型开发流程

Inference Engine 典型的开发流程一共有八步，每一步都对应着相应的 C++ API，每一步都对应着相应的 C++ API，如图 7 所示。

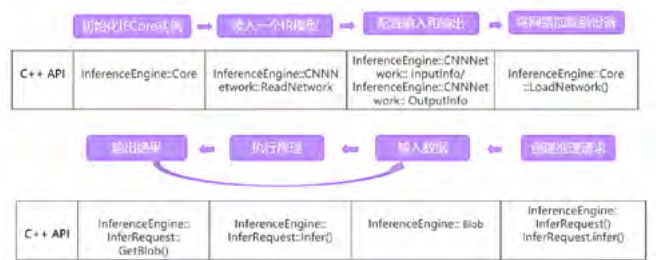


图 7 Inference Engine 典型开发流程

第三步，编写 OpenVINO™ AI 推理计算 C++ 范例

根据第二步的八个 Inference Engine 典型开发程序所对应的 C++ API 即可编写 AI 推理计算程序。

首先完成 Visual Studio 项目属性和环境变量的配置，然后使用 C++ 编程语言开发 OpenVINO™ 推理应用程序，在执行 main() 函数之前引入相应库函数，配置推理计算设备，IR 文件路径，媒体文件路径，部分代码如代码清单 2 所示：

代码清单 2 AI 推理范例部分代码

```
// ----- Step 3. 配置模型输入 & 输出 -----
// -----
// 参考资料: https://docs.openvino toolkit.org/latest/open
vino_docs_IE_DG_Integrate_with_customer_application_
new_API.html
std::cout << "Step 3. Configure input & output." <<
std::endl;
// 获得网络输入信息 (键值对)
InferenceEngine::InputsDataMap inputs_info = network.
getInputsInfo();
// 获得网络输出信息 (键值对)
InferenceEngine::OutputsDataMap output_info = network.
getOutputsInfo();

std::string image_info_name = "";
// 存储 image_info 的名字
std::string input_tensor_name = "";
// 存储 input_tensor 的名字
auto item = inputs_info.begin();
image_info_name = item->first;
// 获取 image_info 输入的名字
auto image_info_ptr = item->second;
// 获取 image_info 输入的指针
item++;
input_tensor_name = item->first;
// 获取 input_tensor 输入的名字
auto input_tensor_ptr = item->second;
// 获取 input_tensor 输入的指针
std::cout << "image_info_name:" << image_info_name << ";
```

```
input_tensor_name:" << input_tensor_name << std::endl;
```

```
// 配置 input_tensor 输入: U8,NCHW, 保持默认的禁止自动  
放缩输入图像和禁止自动转换颜色通道
```

```
input_tensor_ptr->setPrecision(InferenceEngine::Precision::U8);
```

```
// U8 最通用, 参考资料: https://docs.openvino-toolkit.org/  
latest/openvino_docs_IE_DG_supported_plugins_  
Supported_Devices.html
```

```
input_tensor_ptr->setLayout(InferenceEngine::Layout::NC  
HW);
```

编译好程序后, 将解决方案配置设置为 Release 和 X64, 单击本地 windows 调试器”编译并运行 cats_dogs 程序, 编译结果如图 8 所示:

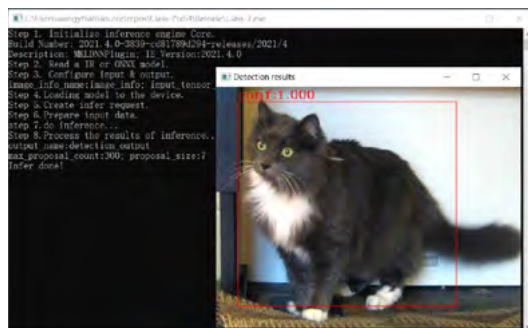


图 8 AI 范例程序编译结果

四、总结

本文通过从零开始训练猫狗数据集目标检测模型, 详细介绍了基于 TensorFlow Object Detection API 框架和预训练模型, 使用 Python 编程语言训练猫狗 FasterRCNN 目标检测模型的完整流程, 在模型部署模块中, 详细介绍了 OpenVINO™ 工具套件及其两个重要组件: Model Optimizer 和 Inference Engine 的安装和使用, 将训练完毕的 FasterRCNN 模型转换为 IR 格式文件后, 使用 C++ 编程语言进行 AI 应用程序的开发方法。在使用 Inference Engine 进行模型推理部署是, OpenVINO™ 工具套件提供了八个统一的 API 接口, 流程化的开发方式极大降低了 AI 应用程序开发的难度, 对于不同模型的不同输入输出节点, 只需少量改动应用程序, 便可快速独立开发属于自己的 AI 应用程序。

详细步骤和完整代码见链接:

如欲了解更多 OpenVINO™ 开发资料,

请扫描下方二维码, 我们会把最新资料及时推送给您。



请访问 [www.Intel.com/PerformanceIndex](https://www.intel.com/PerformanceIndex) 了解负载及参数。结果可能不同。

性能结果基于截至配置中显示的日期的测试, 可能无法反映所有公开可用的更新。有关配置的详细信息, 请参见备份。没有任何产品或组件能够做到绝对安全。

成本及结果均不同。

英特尔技术可能需要支持的硬件、软件或服务得以激活。

英特尔并不控制或审计第三方数据。请您咨询其他来源, 并确认提及数据是否准确。

© 英特尔公司。英特尔、英特尔标识以及其他英特尔商标是英特尔公司或其子公司在美国和/或其他国家的商标。