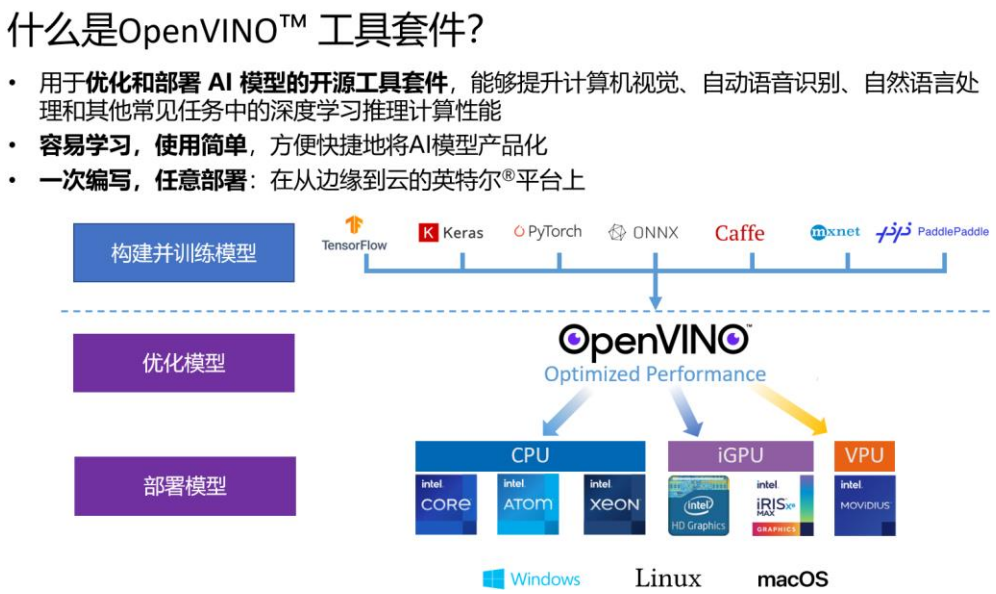
# 用 OpenVINO™ 实现 PyTorch 版 FastSeg 模型高性能推理计算

作者：Boguszewski, Adrian adrian.boguszewski@intel.com

翻译&校对：张晶

## 简介：

PyTorch(https://pytorch.org/)是当今非常流行的，能加速从研究原型设计到生产部署的开源机器学习框架。OpenVINO™ 工具套件能转换并优化深度学习模型，提升深度学习模型在英特尔®计算硬件上的推理计算性能。




使用 OpenVINO™ 工具套件实现 PyTorch 版 FastSeg 模型推理计算，可分为三步：

- 第一步：安装 OpenVINO™ 工具套件
- 第二步：转换 FastSeg 模型
- 第三步：实现 FastSeg 模型推理程序

本文将依次介绍。

## 安装 OpenVINO™ 工具套件

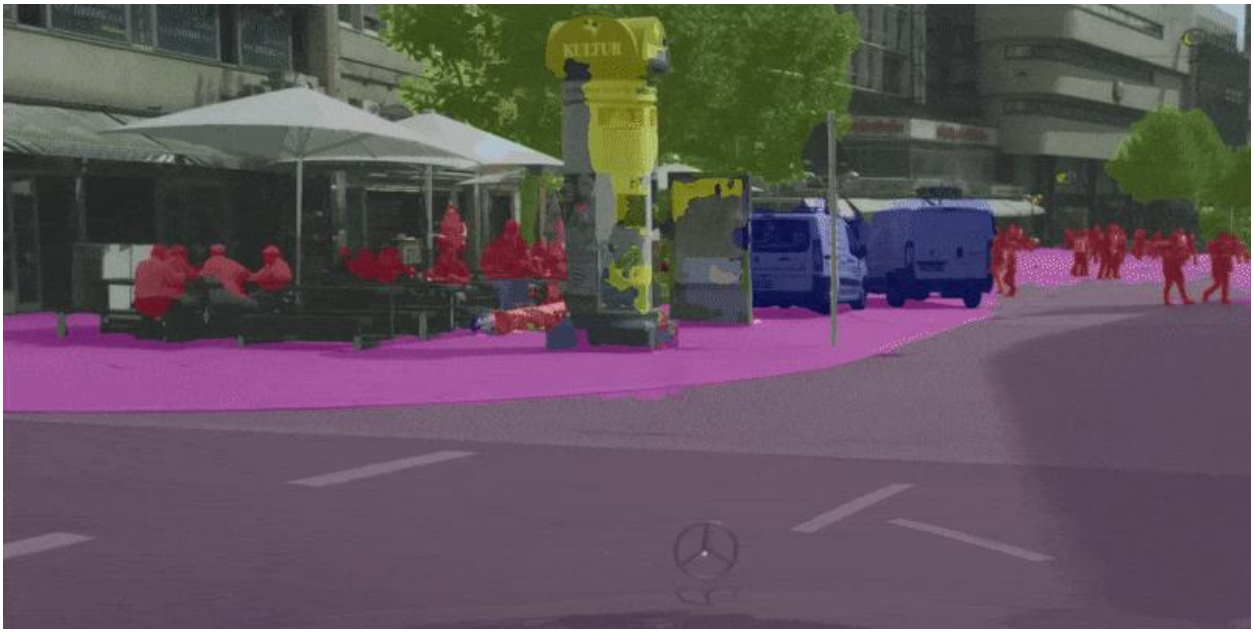
安装 OpenVINO™ 工具套件，非常简单，只需要一条命令：

```
pip install openvino-dev[onnx]
```

详情参考：https://pypi.org/project/openvino-dev/

## 安装 FastSeg 模型

FastSeg 是一个实时语义分割模型，并自带在 Cityscapes(https://www.cityscapes-dataset.com/)数据集上的预训练权重，可用于对各种真实世界街道图像进行有效分割。



FastSeg 模型运行效果

FastSeg 模型使用非常简单，只需要一条命令：

```
pip install fastseg
```

详情参考：https://pypi.org/project/fastseg/

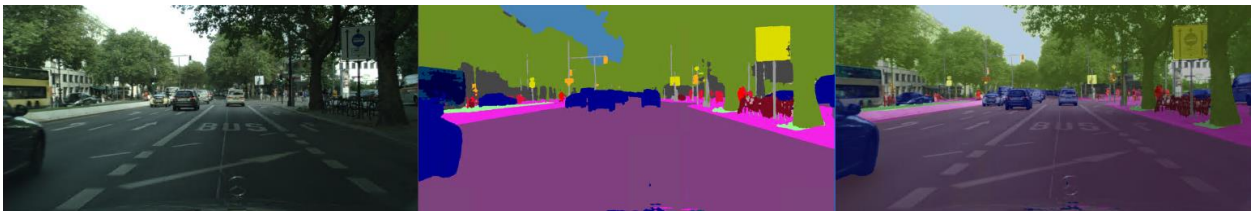
GitHub 代码仓：https://github.com/ekzhang/fastseg

安装完毕后，运行测试代码：https://gitee.com/ppov-nuc/fastseg_openvino_infer/blob/master/fastseg_demo.py

```
from PIL import Image
from fastseg import MobileV3Large
from fastseg.image import colorize, blend

# Load model with pretrained weights
model = MobileV3Large.from_pretrained().cpu().eval()

# Check the original image
img = Image.open('street.png')
```

```
img.resize((800, 400)).show()

# Run the inference and show the labels
labels = model.predict_one(img)
print('Shape:', labels.shape)
print(labels)

# Display the colorized labels
colorized = colorize(labels)
colorized.resize((800, 400)).show()

# Display the composited pic
composited = blend(img, colorized)
composited.resize((800, 400)).show()
```

可以得到模型运行结果如下。



## 导出 FastSeg 模型到 ONNX 格式

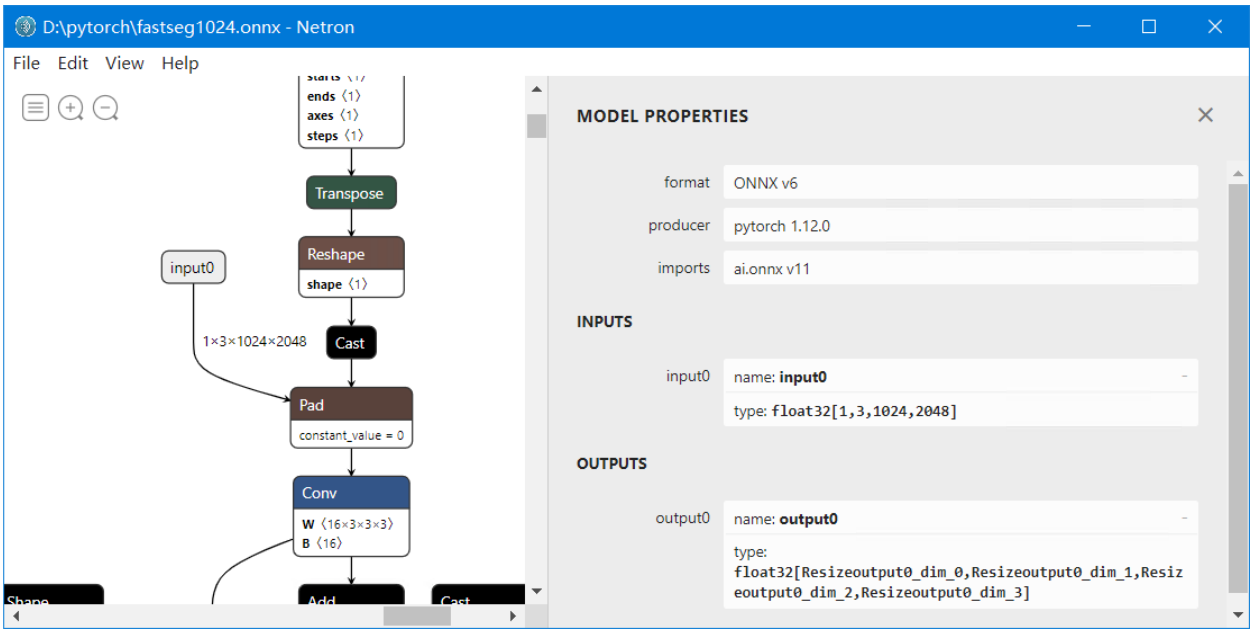
OpenVINO™ 2022.1 版还不能直接读取.pt 格式的模型，所以需要将 FastSeg 的 PyTorch 格式模型先导出为 ONNX 格式模型。

具体导出方式，如下面所示，代码链接：https://gitee.com/ppov-nuc/fastseg_openvino_infer/blob/master/fastseg_export_onnx.py

```
import torch
import geffnet
from fastseg import MobileV3Large

geffnet.config.set_exportable(True)
model = MobileV3Large.from_pretrained().eval()

dummy_input = torch.randn(1, 3, 1024, 2048)
input_names = ['input0']
output_names = ['output0']

model(dummy_input)
```

```
print("Export PyTorch Fastseg model")

torch.onnx.export(model, dummy_input, "fastseg1024.onnx", verbose=False,
    input_names=input_names, output_names=output_names,
    opset_version=11, keep_initializers_as_inputs=True,
do_constant_folding=False)
```

得到 `fastseg1024.onnx` 模型后，可以用 Netron 软件查看，如下图所示。




## 将 ONNX 模型转换为 OpenVINO™ IR 模型

使用命令:

```
mo --input_model fastseg1024.onnx --input_shape "[1,3,1024,2048]" --
mean_values="[123.675,116.28,103.53]" --scale_values="[58.395,57.12,57.375]"
--data_type FP16
```

即可将 `fastseg1024.onnx` 模型转换为 IR 模型: fastseg1024.xml + fastseg1024.bin

## 开发 OpenVINO™ 推理程序

基于 IR 模型: fastseg1024.xml + fastseg1024.bin 的**完整** OpenVINO™ 推理程序，如下所示。代码链接：https://gitee.com/ppov-nuc/fastseg_openvino_infer/blob/master/fastseg_ov_infer.py

```
import cv2
import numpy as np
```

```
from openvino.runtime import Core
from fastseg.image import colorize, blend

image_filename = "street.png"
image = cv2.cvtColor(cv2.imread(image_filename), cv2.COLOR_BGR2RGB)

resized_image = cv2.resize(image, (2048, 1024))

# Convert the resized images to network input shape [1,3,1024,2048]
input_image = np.expand_dims(np.transpose(resized_image, (2, 0, 1)), 0)

# Load the network in Inference Engine
core = Core()
model_ir = core.read_model(model="fastseg1024.xml")
compiled_model_ir = core.compile_model(model=model_ir, device_name="CPU")

# Get output layer
output_layer_ir = compiled_model_ir.output(0)

# Run inference on the input image
res_ir = compiled_model_ir([input_image])[output_layer_ir]
result_mask_ir = np.squeeze(np.argmax(res_ir, axis=1)).astype(np.uint8)

# Show the result
colorized = colorize(result_mask_ir)
colorized.show()

from PIL import Image
img = Image.open(image_filename)
composited = blend(img, colorized)
composited.show()
```

本代码所用测试图片下载链接：https://gitee.com/ppov-nuc/fastseg_openvino_infer/blob/master/street.png

运行结果，如下图所示。


本文对应的 OpenVINO™ Notebook 范例代码：https://github.com/openvinotoolkit/openvino_notebooks/tree/main/notebooks/102-pytorch-onnx-to-openvino

你还在等什么，快来下载源代码，在自己的个人电脑上尝试一下吧！

关于英特尔 OpenVINO™ 开源工具套件的详细资料，包括其中我们提供的三百多个通过验证并优化的预训练模型的详细资料，请您点击 https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/overview.html

为了方便大家了解并快速掌握 OpenVINO™ 的使用，我们还提供了一系列开源的 Jupyter notebook demo。运行这些 notebook，就能快速了解在不同场景下如何利用 OpenVINO™ 实现一系列，包括 OCR 在内的，计算机视觉及自然语言处理任务。OpenVINO™ notebooks 的资源可以在 Github 这里下载安装：https://github.com/openvinotoolkit/openvino_notebooks