

用 AI 及 OpenVINO™ 寻找失落的侏罗纪

作者： 武卓

在世界各地，当你走进博物馆，看到巨大的恐龙标本展示，你有没有想过古生物学家是如何发现它们的？

有趣的是，我们的一位好朋友兼同事，英特尔的技术布道师 Bob Chesebrough，在闲暇时间和他的儿子们一起出去寻找尚未被发现的恐龙化石。他向我们展示了如何做到这一点，以及我们如何使用 AI 和 OpenVINO™ 来明显简化这一过程。

鲍勃向我们解释说，古生物学家和考古学家必须首先找到这些地点，其地质历史不仅揭示了化石的创造过程，而且还表明了发现化石的机会。传统上，这是通过扫描航拍图像来发现特定的地形纹理，或岩层的颜色和方向，然后构建像这样的“可能性地图”来完成（如图 1 所示）。

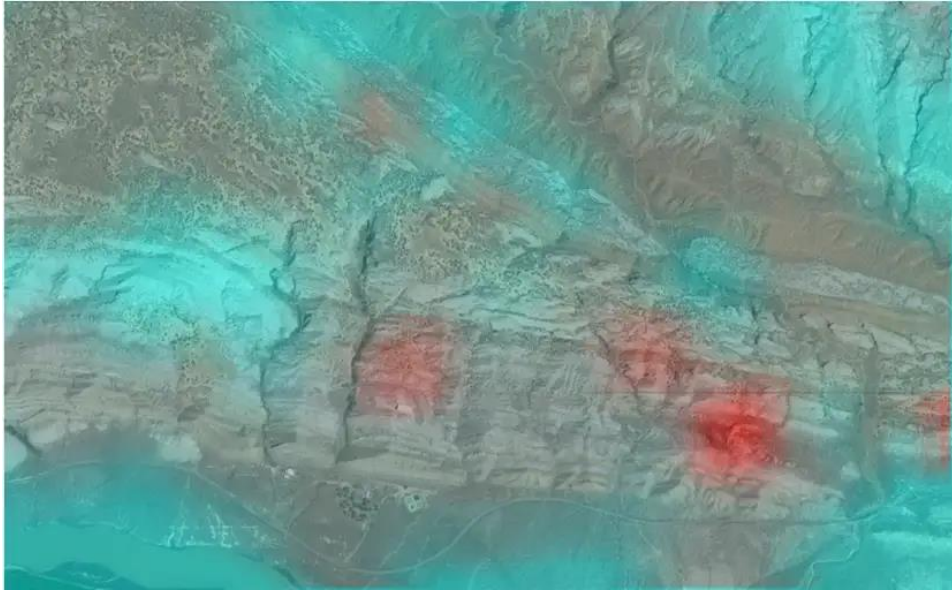


图 1. 恐龙化石可能性地图示例

你可以在图片中看到可能发现恐龙化石的所有位置。你可以想象的是，即使是由专家亲自完成，绘制这样一张地图也需要花费大量的时间和精力。但是，由于 Bob 已经了解了 AI 的力量，他想知道是否可以开发一种能够自动创建这些地图的 AI 模型。他做到了！

在他的最初尝试中，他发现 AI 软件运行良好，但在野外创建地图花费了太长时间，而你在野外并依赖笔记本电脑电池时，这是一个问题。然后，他想知道如果他使用无人机拍摄高分辨率航拍照片是否会有所帮助，但不幸的是，这也行不通，因为这仍然需要他回家上传这些照片。

因此，他与我们（Raymond Lo 和 武卓 博士）以及英特尔的其他 AI 布道师一起提出了一个解决方案，使他能够在几分钟内（通常这需要几个小时）从任意组合的视频中创建这样的可能性地图。

我们很高兴向您展示我们做了什么以及如何使用 OpenVINO，但在我们继续之前，请注意，虽然狩猎恐龙化石听起来是一种有趣和令人兴奋的爱好的，但在美国 and 大多数其它地方，恐龙化石的实际采集受到严格监管，如果不是完全禁止的话。因此，如果你亲自尝试，而且你确实发现了化石的话，我们强烈建议你拍照并向当地有关部门报告你的这一发现。

狩猎开始了！

现在让我们开始吧！首先，我们将解释寻找恐龙化石的不同阶段，然后我们将深入了解具体的步骤是怎么做的。

[具体的创建可能性地图的步骤可以参考这里](#)（如图 2 所示），包括：

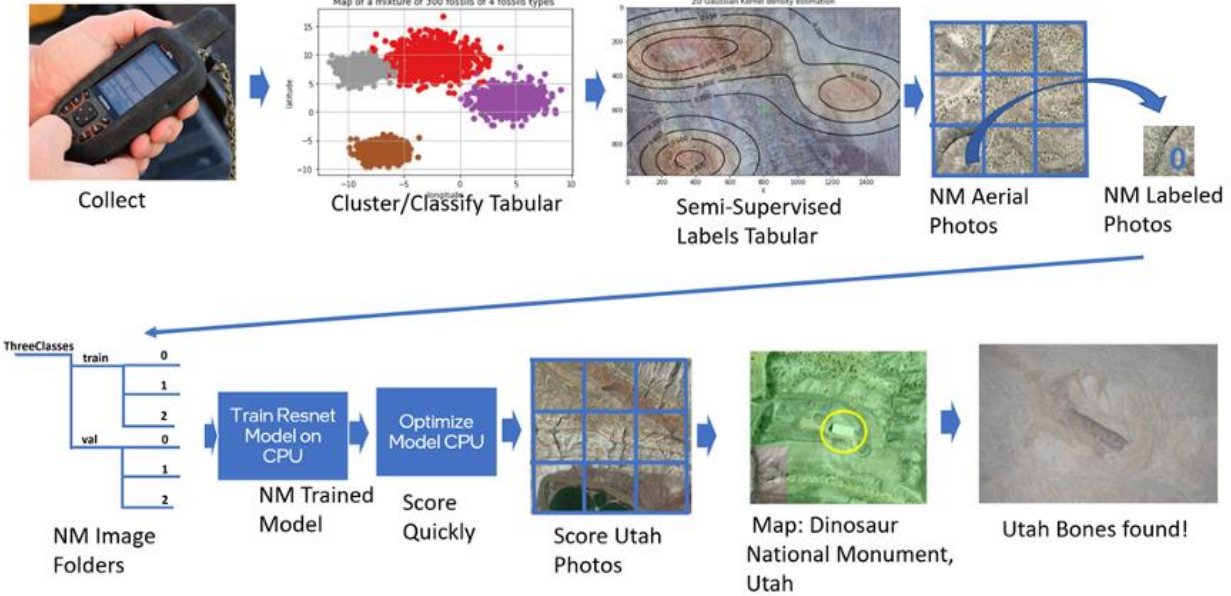


图 2. 创建恐龙化石可能性地图的一般步骤

第一阶段和第二阶段：收集数据并对其进行分类，包括获取已知恐龙骨床（示例中为新墨西哥州）化石的实际发现的 GPS 坐标，并根据它们之间的接近程度对它们进行聚类。我们使用 Intel® 机器学习库 Scikit learn 做到了这一点。

第三阶段：标记数据。我们将聚类的结果和 Intel® Deep Neural Network Library 应用于标记为“无化石”、“极有可能找到化石”或“可能找到化石”的相同位置的航拍图像，并将这些图像合并起来，得到该区域的化石可能性地图。

第四阶段和第五阶段：训练数据并通过将标记的航拍图像转换为张量来构建模型，每个张量都具有与概率分数或置信水平相对应的类名（例如，0 表示“绝对没有化石”，1 表示“有可能找到化石”，2 表示“发现化石”），以使用 PyTorch 构建残差神经网络（ResNet）。

第六阶段：最后，评分阶段，通过使用前面步骤中准备的模型，利用 OpenVINO 推理引擎分析航拍照片，创建其它地方的化石可能性地图！

放“码”过来！

实现上述过程的代码由一系列 [Jupyter Notebook](#) 提供。这里，我们描述了最重要的部分，如图 3 所示，将 PyTorch 模型转换为 OpenVINO 中间表示（IR）格式并分析图像。

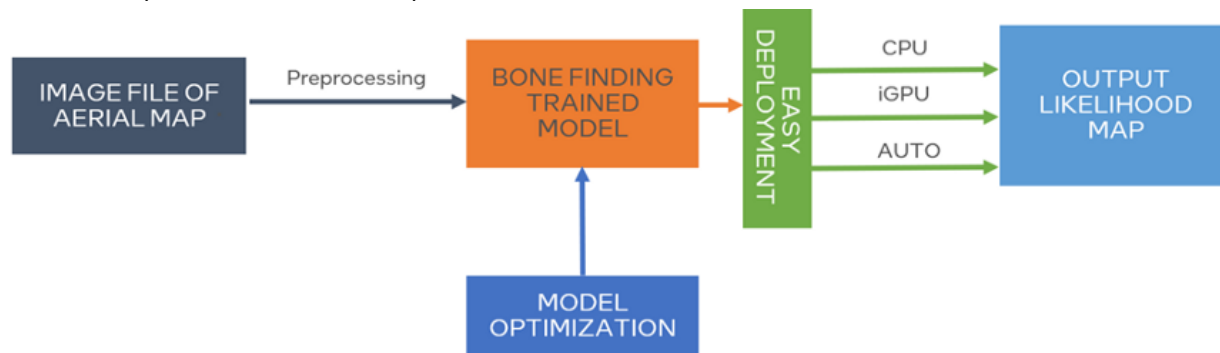


图 3. PyTorch 模型转换为 OpenVINO™ IR 格式

加载 OpenVINO 推理引擎只需要一条指令：

```
from openvino.runtime import Core
```

加载所有必要的库后，Notebook 定义了先前以 PyTorch 格式训练得到的化石寻找模型的路径，并定义了两个新模型的位置，以创建一个以 ONNX 格式，另一个以 OpenVINO IR 格式的模型，原因我们稍后将解释。

```
DIRECTORY_NAME = "models"

BASE_MODEL_NAME = DIRECTORY_NAME +
f"/bc_resnet18_simple_NOIPEX_6Epochs_StateDict_gold"

# Paths where PyTorch, ONNX and OpenVINO IR models will be stored
model_path = Path(BASE_MODEL_NAME).with_suffix(".pt")
onnx_path = model_path.with_suffix(".onnx")
ir_path = model_path.with_suffix(".xml")
```

生成 ONNX 模型后，再使用 OpenVINO 模型优化器（MO）构建命令，将其转换为 FP16 精度的 OpenVINO IR 格式：

```
# Construct the command for Model Optimizer
mo_command = f"""mo
    --input_model "{onnx_path}"
    --input_shape "[1,3, {IMAGE_HEIGHT}, {IMAGE_WIDTH}]"
    --data_type FP16
    --output_dir "{model_path.parent}"
    """
mo_command = " ".join(mo_command.split())
```

在使用 ONNX 模型对测试图像运行推理（此处未显示）后，在同一图像上重复该过程，但使用 OpenVINO IR 格式模型，因为在 OpenVINO 推理引擎上执行两个模型显示了如何在 CPU 或 GPU 设备上可互换地运行模型推理。

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

# Get input and output layers
output_layer_ir = compiled_model_ir.output(0)

# Run inference on the input image
res_ir = compiled_model_ir([x_test2])[output_layer_ir]
```

上面的前几行代码加载 OpenVINO 推理引擎，将 OpenVINO IR 模型传递给它，然后提取其输入和输出层。最后一行代码是在先前使用 ONNX 模型分析的另一图像（“x_test2”）上实际运行推理引擎的代码。

接下来，程序打印出模型推理的结果；即该图像所属的类（0、1 或 2），然后继续使用 PyTorch 模型重复该过程。

代码的最后部分做了两件事：

- 1) 它运行 OpenVINO 和 PyTorch 模型，对几百张图像进行分类，并对所使用的三种模型的性能进行近似比较。
- 2) 它将所有这些图像合并到一张地图中，如图 1 所示，并通过将 PyTorch 结果与 OpenVINO 结果进行比较得出结论，以验证准确性没有损失。为了简洁起见，这里我们只包含一段不言自明的片段：

```
if "GPU" in core.available_devices:
    num_images = 2000

compiled_model_onnx_gpu = core.compile_model(model=model_onnx,
device_name="GPU")
#warm up
compiled_model_onnx_gpu([input_image])
```

使用 OpenVINO 的好处不仅在于我们可以获得加速，还在于能够根据您的硬件优化模型，并充分利用您手边可用的硬件！

后续步骤

整个的步骤就是这样！现在你是一个现代寻宝者，能够使用 OpenVINO 为您的寻找侏罗纪之旅创建和更新可能性地图。

这是一个很好的例子，说明 OpenVINO 如何让每个人无论做什么都能更容易地使用人工智能。

您可以轻松地将此方法用于任何需要从航拍照片中检测某些区域的情况，例如防火、可再生能源发电等关键领域，甚至其它星球上的生命！

如果您想了解更多信息，我邀请您观看我的完整演示，或 [Bob 的演示](#)，他将指导您下一次的恐龙化石寻找！要了解更多关于如何构建自己的 AI 应用程序的信息，请查看我们的 [OpenVINO 开发人员资源](#)并立即开始。

通知和免责声明

Intel 技术可能需要启用硬件、软件或服务激活。

任何产品或组件都不能绝对安全。

您的成本和结果可能有所不同。

©英特尔公司。Intel、Intel 徽标和其他 Intel 标志是 Intel Corporation 或其子公司的商标。其他名称和品牌可能被称为他人的财产。