

AI と OpenVINO™ を使用して恐竜の骨を発掘する

この記事は、Medium に公開されている「[Using AI and OpenVINO™ to Dig Up Dinosaur Bones](#)」の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

著者: Robert Chesebrough インテル コーポレーション oneAPI AI テクニカル・エバンジェリスト
Raymond Lo インテル コーポレーション AI ソフトウェア・エバンジェリスト
Zhuo Wu インテル コーポレーション AI ソフトウェア・エバンジェリスト



世界中の博物館には巨大な恐竜の標本が展示されていますが、古生物学者がどのようにして恐竜の標本を発見しているのか、立ち止まって疑問に思われたことはありませんか？

面白いことに、我々の友人であり同僚でもあるテクニカル・エバンジェリストの Bob Chesebrough は、時間があるときに子供たちと一緒に恐竜の骨を実際に探しに出かけています。そして、この作業に取り組むために必要なこと、AI と OpenVINO™ を使用してプロセスを劇的に簡素化する方法を的確に示してくれました。

Bob は、古生物学者と考古学者はまず、化石が生成されるだけでなく、化石を発見できる可能性がある地質年代の場所を見つける必要があると説明してくれました。従来、この作業は、空中写真をスキャンし、特定の地形テクスチャーや岩層の色と方向を見分けて、「尤度 (可能性) マップ」(図 1) を作成することにより行われていました。

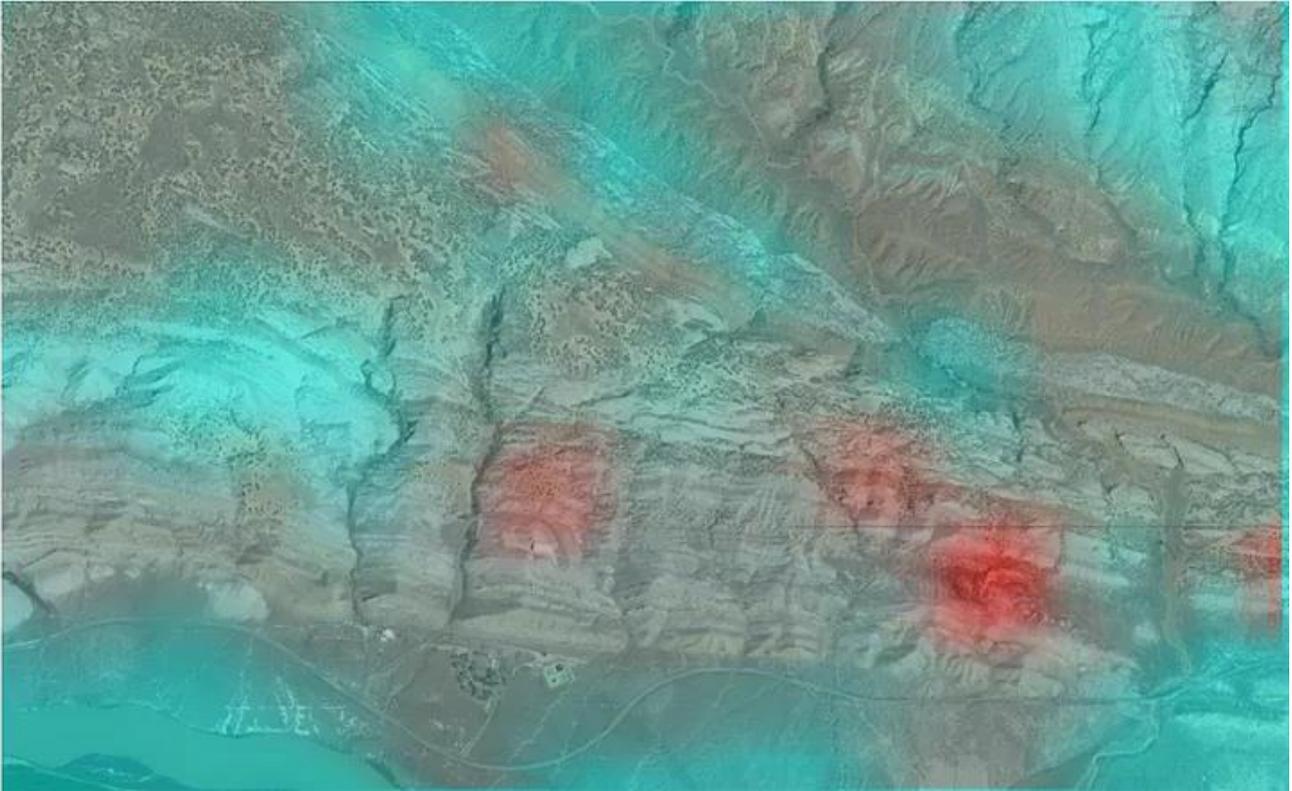


図 1. 恐竜の骨の尤度マップの例

尤度マップを見れば、恐竜の骨が見つかりやすい場所が分かります。しかし、マップを作成する作業は、たとえ専門家でも、手作業で行うと大変な時間と労力がかかります。Bob は AI の能力を知っていたため、これらのマップを自動的に作成する AI モデルを開発できるのではと考えました。そして、開発に成功したのです。

最初に現地で試したとき、ソフトウェアは適切に動作しましたが、マップの作成に非常に時間がかかったため、ノートブックのバッテリーが切れてしまいました。次に、ドローンを使って高解像度の空中写真を撮影する方法を考えましたが、残念ながら、写真をアップロードするには自宅に戻る必要があったため、この方法も採用されませんでした。

そこで彼は、我々およびインテルのほかの AI エバンジェリストと協力して、[任意の映像を組み合わせることで尤度マップを数分で作成するソリューション](#) (英語) を考え出しました (通常は、マップの作成には数時間かかります)。

今から我々が何をしたのか、OpenVINO™ をどのように使用したのかを紹介しますが、その前に注意点を忘れなく。恐竜の骨を探すのは楽しい刺激的な趣味のように聞こえるかもしれませんが、ほとんどの国や地域では、恐竜の骨を実際に収集することは厳しく規制されています。その国や地域のルールを必ず守って行動してください。この記事を読んで自分で試し、実際に化石を発見した場合は、まず写真を撮って当局 (官庁など) に報告することを強く推奨します。

骨探し開始!

では始めましょう! 最初に、主な手順について説明します。

一般的な手順を図 2 に示します。[インテル® AI ツールで恐竜を見つける手順](#)も参照してください。

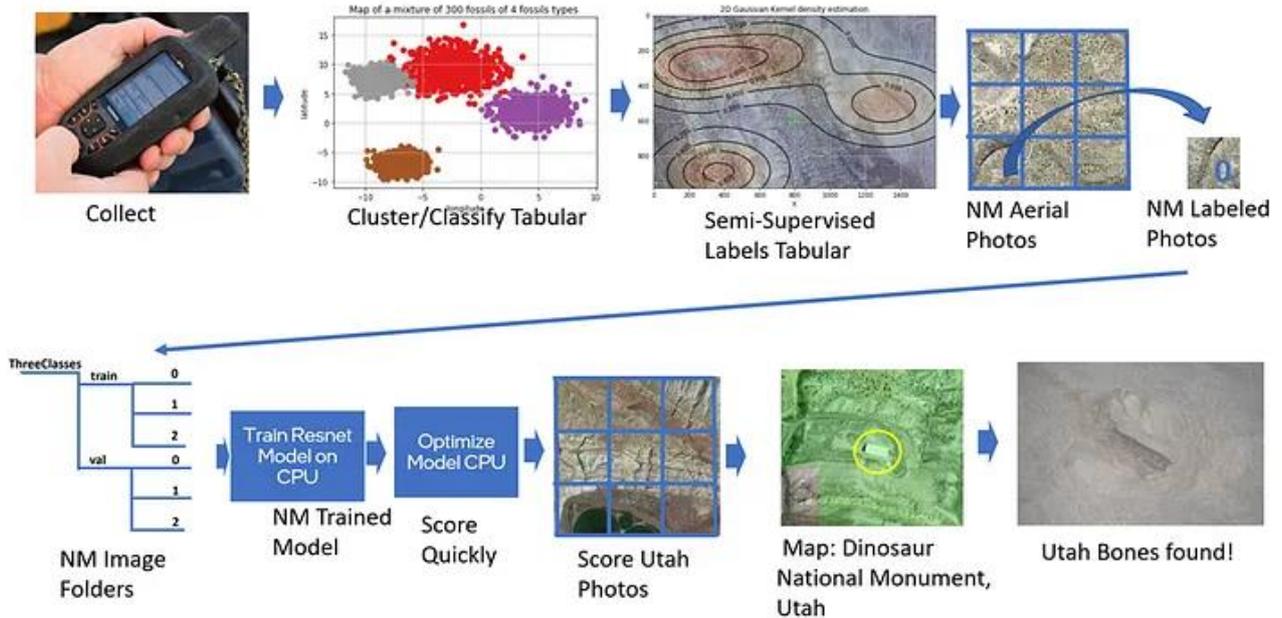


図 2: 骨の尤度マップを作成する一般的な手順

手順 1 および 2: データを収集して分類します。既知の恐竜のボーンベッド (この例ではニューメキシコ州) で実際に発見された化石の GPS 座標の取得と、相互の近接性に基づくクラスタリングが含まれます。作業には、マシンラーニング・ライブラリーの [scikit-learn*](#) [向けインテル® エクステンション](#) (英語) を使用しました。

手順 3: データにラベルを付けます。クラスターと [インテル® oneAPI ディープ・ニューラル・ネットワーク・ライブラリー](#) (英語) を利用し、「no bone (骨なし)」、「bone very likely (骨の可能性が非常に高い)」、「bone possible (骨の可能性あり)」とラベルを付けた同じ場所の空中写真に適用して、その領域の骨の尤度マップを作成します。

手順 4 および 5: ラベルを付けた空中写真をテンソルに変換し、データをトレーニングしてモデルを構築します。各テンソルに、確率スコアまたは信頼水準 (例: 「骨なし」は 0、「骨の可能性あり」は 1、「骨を発見」は 2) に対応するクラス名を付けます。[PyTorch*](#) (英語) を使用して、ResNet (Residual Neural Network) を構築します。

手順 6: 前の手順で準備したモデルを使用して、OpenVINO™ 推論エンジンで空中写真を分析し、ほかの場所の骨の尤度マップを作成します。

コードの説明

上記の手順の実装方法は、[Jupyter* Notebook のコード](#) (英語) を参照してください。ここでは、最も重要な、PyTorch* モデルを OpenVINO™ 中間表現 (IR) 形式に変換して画像を分析する部分 (図 3) を説明します。

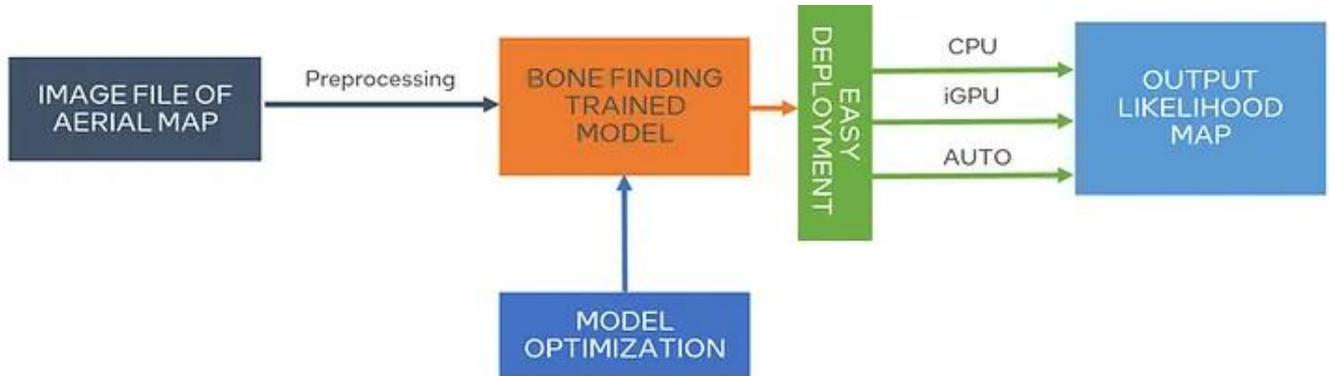


図 3: Pytorch* モデルを OpenVINO™ IR 形式に変換

次の 1 つの命令で、OpenVINO™ 推論エンジンをロードします。

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

必要なすべてのライブラリーをロードした後、以前 PyTorch* 形式で作成したトレーニング済みの骨発見モデルのパスと 2 つの新しいモデルの場所を定義して、1 つを ONNX 形式、もう 1 つを OpenVINO™ IR 形式で作成します。この理由は後で説明します。

```
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")
```

ONNX モデルを作成した後、[OpenVINO™ ツールキットのモデル・オブティマイザー](#) (英語) を使用して、FP16 精度で OpenVINO™ IR 形式に変換するコマンドを構築します。

```
# 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())
```

ONNX モデルを使用してテスト画像で推論を実行 (ここでは示していません) した後、OpenVINO™ モデルを使用して同じ画像で手順を繰り返します。両方のモデルを OpenVINO™ 推論エンジンで実行すると、CPU または GPU デバイスでモデルの推論を実行する方法が示されます。

```
# 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]
```

上記のコードの最初の数行は、OpenVINO™ 推論エンジンをロードして OpenVINO™ IR モデルを渡し、入力層と出力層を抽出します。上記のコードの最後の行は、ONNX モデルで以前分析した同じ画像 (「x_test2」) に推論を実行します。

次に、その画像が属するクラス (0、1、または 2) を推論の結果として出力します。その後、PyTorch* モデルを使用して同じ手順を繰り返します。

コードの最後の部分は、以下の 2 つのことを行います。

1. OpenVINO™ モデルと PyTorch* モデルの両方を実行し、数百の画像を分類して、使用したモデルのパフォーマンスを比較します。
2. すべての画像を図 1 のような 1 つのマップにマージし、PyTorch* の結果と OpenVINO™ の結果を比較して、精度が損なわれていないことを確認します。簡潔に説明するため、ここでは一目で分かるコードのみ示します。

```
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])
```

OpenVINO™ を使用することの素晴らしい点は、高速化に加えて、選択するハードウェアに応じてモデルを最適化し、利用可能なハードウェアを最大限に活用できることです。

次のステップ

説明は以上です! これで皆さんも、OpenVINO™ を使用して骨探しの尤度マップを作成して更新できる現代のトレジャーハンターの仲間入りです。

これは、OpenVINO™ を利用すれば、どんなことをしている人でも AI を活用できる可能性があるという素晴らしい例です。

この方法は、防災、再生可能発電などの重要な分野や、ほかの惑星の生命体の調査など、空中写真から特定の地域を発見する必要がある、あらゆる状況に簡単に再利用できます。

さらに詳しく知りたい場合は、[恐竜の骨探しのガイドとなる Bob のデモ \(英語\)](#) をご覧ください。独自の AI アプリケーションを構築する方法の詳細は、[OpenVINO™ 開発者リソース \(英語\)](#) を参照してください。

OpenVINO™ ツールキットとは

AI を加速する無償のツールである OpenVINO™ ツールキットは、インテルが無償で提供しているインテル製の CPU や GPU、VPU、FPGA などのパフォーマンスを最大限に活用して、コンピュータービジョン、画像関係をはじめ、自然言語処理や音声処理など、幅広いディープラーニング・モデルで推論を最適化し高速化する推論エンジン/ツールスイートです。

OpenVINO™ ツールキット・ページでは、ツールの概要、利用方法、導入事例、トレーニング、ツール・ダウンロードまでさまざまな情報を提供しています。ぜひ特設サイトにアクセスしてみてください。

<https://www.intel.co.jp/content/www/jp/ja/internet-of-things/openvino-toolkit.html>

法務上の注意書き

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

絶対的なセキュリティを提供できる製品またはコンポーネントはありません。

実際の費用と結果は異なる場合があります。

© Intel Corporation. Intel、インテル、Intel ロゴ、その他のインテルの名称やロゴは、Intel Corporation またはその子会社の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。