Linux* で AI 開発キットと OpenVINO™ を使用してリアルタイム人数カウントを行う方法

この記事は、Medium に公開されている「How To Perform Real-Time People Counting on Linux AI Dev Kit With OpenVINO™」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。

インテリジェントな、自動 AI ソリューション・ベースの人数カウントは、小売店やショッピング・モールで歩行者をモニターして顧客に優れたショッピング体験を提供する、公共交通機関でスケジュールを最適化して混雑に対処する、スマートシティーで公共の安全と都市計画を改善するなど、さまざまなシナリオで非常に役立ちます。

OpenVINO™ のモデルの最適化機能と簡単なデプロイ機能、AI 開発キットが提供する強力な計算機能を組み合わせることにより、最小限の労力でリアルタイム人数カウントを実現できます。

一緒にやってみましょう

幸いなことに、この種のアプリケーションはすでに新しい openvino_build_deploy (英語) リポジトリーに存在していて、オープンソースです (Apache* 2.0 ライセンス (英語))。仕組みと実行方法を理解するため、ステップごとに確認してみましょう。確認には、Ubuntu* 24.04 を使用します。

最初に、すべての前提条件、git と python をインストールすることから始めます。

```
sudo apt install git gcc python3-venv python3-dev
```

次に、リポジトリーのクローンを作成します。

```
git clone https://github.com/openvinotoolkit/openvino_build_deploy.git
```

コードが含まれているディレクトリーに移動します。

```
cd openvino_build_deploy/demos/people_counter_demo
```

さまざまなプロジェクト間の依存関係が混在しないように、仮想環境を作成して有効にします。

```
python3 -m venv venv
source venv/bin/activate
```

準備ができたら、要件をインストールします。

```
python -m pip install --upgrade pip
pip install -r requirements.txt
```

これで環境の準備ができました。アプリを実行する前にコードを見てみましょう。ここでは、コードの最も重要な部分のみ取り上げます。ほかの部分は、main.py (英語) ファイルを確認してください。

人をカウントするには、まず人を検出する必要があります。人の検出には、最先端のモデルの 1 つ、Ultralytics (英語) の YOLOv8 を使用します。鍵となる優れたパフォーマンスの実現には、OpenVINO™ (英語) と NNCF (英語) を使用します。次に、モデル (ここでは YOLOv8 nano) を OpenVINO™ 形式でエクスポートします。モデルを量子化する場合は、`int8=True` を設定します。

```
from ultralytics import YOLO

# create a YOLO object detection model
yolo_model = YOLO("yolov8n")

# export the model to OpenVINO format (FP16 and INT8)
yolo_model.export(format="openvino", dynamic=False, half=True)
yolo_model.export(format="openvino", dynamic=False, half=True, int8=True,
data="coco128.yaml"))
```

OpenVINO™ 形式でエクスポートしたモデルを CPU、GPU、NPU などの特定のデバイス向けにロードしてコンパイルする前に、利用可能なすべてのデバイスを表示します。

```
from openvino import runtime as ov

core = ov.Core()
for device in core.available_devices:
    device_name = core.get_property(device, "FULL_DEVICE_NAME")
    print(device_name)
```

ビデオフレームの結果をできるだけ早く取得できるように、レイテンシー・モードでモデルをコンパイルすることが重要です。

```
# read the model from file
model = core.read_model(model_path)

# compile the model for latency mode
model = core.compile_model(model, device_name="NPU",
config={"PERFORMANCE_HINT": "LATENCY"})
```

推論の前後には前処理 (英語) と後処理 (英語) が必要です。この記事では省略しますが、リンクにアクセスして確認してください。これで、モデルの推論を行う準備ができました。

```
results = model(input_image)
```

わずか 1 行のコードで、入力画像から結果が生成されます。しかし、これらの結果はまだ解釈が必要です。解釈には supervision (英語) ライブラリーを使用します。このライブラリーは、指定された領域内の人をフィルタリング、アノテート、カウントするのに役立ちます。ビデオストリームの下部のアノテーターを作成します。

```
polygon = [[0, 360], [0, 1080], [1920, 1080], [1920, 360]]

# a zone to count people in
zone = sv.PolygonZone(polygon=polygon, frame_resolution_wh=(1920, 1080))

# the annotator — visual part of the zone
zone_annotator = sv.PolygonZoneAnnotator(zone=zone)

# box annotator, showing boxes around people
box_annotator = sv.BoxAnnotator()
```

これらを解釈に使用します。

```
frame = zone_annotator.annotate(scene=frame)

# get detections relevant only for the zone
mask = zone.trigger(detections=detections)
detections_filtered = detections[mask]

# visualize boxes around people in the zone
frame = box_annotator.annotate(scene=frame, detections=detections_filtered)
```

最後に必要なのは、人数をカウントしてアラートを出すことです。これは非常に簡単です。

```
people_count = len(detections_filtered)

# add alert text to the frame if necessary
if people_count > people_limit:
    utils.draw_text(frame, text=f"Intel employee required in zone
{zone_id}!", point=(20, 20), font_color=(0, 0, 255))
```

すべてのコードを実行して結果を確認するには、コマンドラインに次のように入力します。

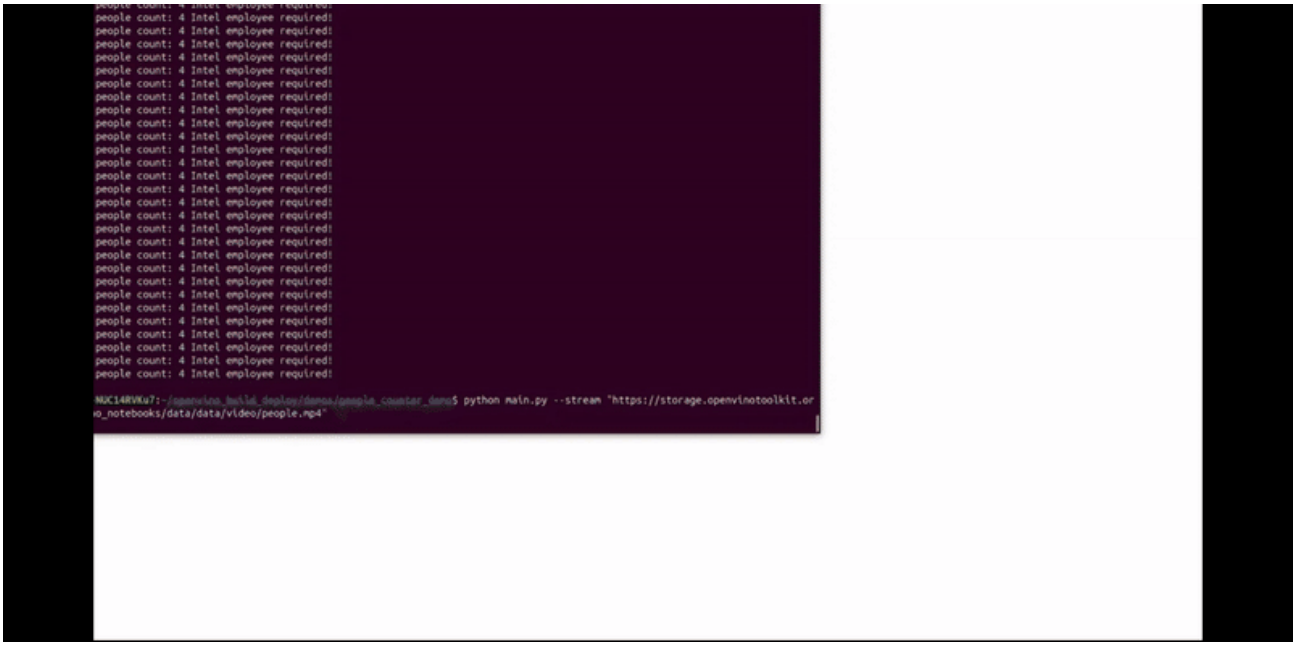
```
python main.py --stream 0
```

ここで、数字 (例: 0) はウェブカメラの番号を示します。代わりにビデオパスを使用することもできます。モデルや people_limit など、変更するパラメーターはほかにもいくつかあります。パラメーターの情報を確認するには、ヘルプオプションを実行します。

```
python main.py --help
```

結果

ビデオパスで人数カウントを実行した結果を次に示します。コントロール・パネルで異なる数字のキーを押すことにより、AI 開発キット内の CPU、GPU、NPU 間の推論を簡単に切り替えることができます。



ローカル AI 開発キットの AI デモを試してみてください。

ほかの AI のデモにも興味がある場合は、GitHub* の openvino_build_deploy(英語) リポジトリーを参照してください。リポジトリーには、実際のアプリケーションで AI の機能を紹介する、すぐに実行可能なデモが用意されています。パフォーマンスの最適化、高度なモデルのデプロイはもちろん、単に新しいアイデアの実験に興味がある場合でも、このコードベースは最適な出発点となります。リポジトリーのクローンを作成して自分のマシンでデモを実行することにより、実践的な経験と洞察を得て、プロジェクトを加速できます。AI の世界に飛び込むこのチャンスを逃さないようにしてください。今すぐリポジトリーのクローンを作成して、AI の旅を始めましょう。

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