

[入门指南](#)  
[操作步骤](#)  
[指南](#)  
[资源](#)  
[性能信息](#)  
[API 参考](#)

[资源概述](#)

[示例](#)

- [图像分类 C++ 示例异步](#)
- [图像分类 Python\\* 示例异步](#)
- [Hello 分类 C++ 示例](#)
- [图像分类 Python\\* 示例](#)
- [Hello Reshape SSD C++ 示例](#)
- [Hello NV12 输入分类 C++ 示例](#)
- [Hello 查询设备 C++ 示例](#)
- [Hello 查询设备 Python\\* 示例](#)
- [对象检测 C++ 示例 SSD](#)
- [对象检测 Python\\* 示例 SSD](#)
- [自动语音识别 C++ 示例](#)
- [神经风格迁移 C++ 示例](#)
- [神经风格迁移 Python\\* 示例](#)
- [性能指标评测 C++ 应用](#)
- [性能指标评测 Python\\* 应用](#)

[演示](#)

- [十字路口摄像头演示](#)
- [交互式面部检测 C++ 演示](#)
- [交互式人脸识别 Python\\* 演示](#)
- [TensorFlow\\* 对象检测 Mask R-CNN 分割演示](#)
- [多通道演示](#)
  - [多通道面部检测 C++ 演示](#)
  - [多通道人体姿态估计 C++ 演示](#)
  - [多通道对象检测 YOLO\\* V3 C++ 演示](#)
- [使用 Faster R-CNN 的对象检测演示](#)
- [面向 CenterNet Python\\* 的对象检测演示](#)
- [对象检测 SSD C++ 演示，异步 API 性能展示](#)
- [对象检测 SSD Python\\* 演示，异步 API 性能展示](#)
- [安全障碍摄像头演示](#)

- 图像分割 C++ 演示
- 单个人体姿态估计 Python\* 演示
- 图像分割 Python\* 演示
- 智能课堂演示
- 文本检测演示
- 文本辨认 Python\* 演示
- 视线预估演示
- 人体姿态估计演示
- 3D 人体姿态估计 Python\* 演示
- 行人追踪器演示
- 超高分辨率演示
- 对象检测 YOLO\* V3 C++ 演示, 异步 API 性能展示
- 对象检测 YOLO\* V3 Python\* 演示, 异步 API 性能展示
- 动作识别演示
- 实例分割演示
- 3D 分割演示
- 图像检索 Python\* 演示
- 多通道多人追踪 Python\* 演示
- 语音库与语音识别演示
  - 语音库
  - 离线语音识别演示
  - 现场语音识别演示
  - Kaldi\* 统计语言模型转换工具

## 工具

- 精度检查器工具
  - 精度检查器示例
  - 配置 Caffe\* 启动程序
  - 配置 OpenVINO 启动程序
  - 配置 OpenCV\* 启动程序
  - 配置 MxNet\* 启动程序
  - 配置 TensorFlow\* 启动程序
  - 配置 TensorFlow\* Lite 启动程序
  - 配置 ONNX\* Runtime 启动程序
  - 配置 \*PyTorch 启动程序
  - 适配器
  - 注释转换器
  - 预处理程序
  - 后处理程序
  - 指标
  - 面向精度检查器的定制评估器
  - 阅读器
  - Caffe\* 安装技巧

训练后优化

量化

[DefaultQuantization](#) 算法

[AccuracyAwareQuantization](#) 算法

[使用训练后优化工具套件 API](#)

[定义配置文件](#)

[配置文件示例](#)

模型下载器

[交叉检查工具](#)

[编译工具](#)

预训练模型 ( 开放式 Model Zoo )

对象检测模型

[face-detection-adas-0001](#)

[face-detection-adas-binary-0001](#)

[face-detection-retail-0004](#)

[face-detection-retail-0005](#)

[person-detection-retail-0002](#)

[person-detection-retail-0013](#)

[person-detection-action-recognition-0005](#)

[person-detection-action-recognition-0006](#)

[person-detection-action-recognition-teacher-0002](#)

[person-detection-raisinghand-recognition-0001](#)

[person-detection-asl-0001](#)

[pedestrian-detection-adas-0002](#)

[pedestrian-detection-adas-binary-0001](#)

[pedestrian-and-vehicle-detector-adas-0001](#)

[vehicle-detection-adas-0002](#)

[vehicle-detection-adas-binary-0001](#)

[person-vehicle-bike-detection-crossroad-0078](#)

[person-vehicle-bike-detection-crossroad-1016](#)

[product-detection-0001](#)

[vehicle-license-plate-detection-barrier-0106](#)

对象识别模型

[age-gender-recognition-retail-0013](#)

[head-pose-estimation-adas-0001](#)

[asl-recognition-0003](#)

[license-plate-recognition-barrier-0001](#)

[vehicle-attributes-recognition-barrier-0039](#)

[emotions-recognition-retail-0003](#)

[landmarks-regression-retail-0009](#)

facial-landmarks-35-adas-0002  
person-attributes-recognition-crossroad-0230  
gaze-estimation-adas-0002

#### 重识别模型

person-reidentification-retail-0031  
person-reidentification-retail-0103  
person-reidentification-retail-0107  
person-reidentification-retail-0200  
face-reidentification-retail-0095

#### 语义分割模型

road-segmentation-adas-0001  
semantic-segmentation-adas-0001

#### 实例分割模型

instance-segmentation-security-0050  
instance-segmentation-security-0083  
instance-segmentation-security-0010

#### 人体姿态估计模型

human-pose-estimation-0001

#### 图像处理

single-image-super-resolution-1032  
single-image-super-resolution-1033  
text-image-super-resolution-0001

#### 文本检测

text-detection-0003  
text-detection-0004

#### 文本识别

text-recognition-0012  
handwritten-score-recognition-0003

#### 文本辨认

text-spotting-0001-detector  
text-spotting-0001-recognizer-decoder  
text-spotting-0001-recognizer-encoder

#### 动作识别模型

driver-action-recognition-adas-0002-encoder  
driver-action-recognition-adas-0002-decoder  
action-recognition-0001-encoder  
action-recognition-0001-decoder

#### 图像检索

image-retrieval-0001

#### 压缩模型

resnet18-xnor-binary-onnx-0001  
resnet50-binary-0001

## 人体姿态估计 C++ 演示

### 本文档

[操作步骤](#)

[运行](#)

[演示输出](#)

[另请参阅](#)

该演示展示了多人 2D 姿态估计算法的操作步骤。任务是预测输入视频中每个人的姿态：人体骨骼，包含关键点和关键点之间的连接。姿态可能包含多达 18 个关键点：*耳朵、眼睛、鼻子、颈部、肩部、肘部、腕部、臀部、膝盖和脚踝*。该算法的潜在用例包括动作识别和行为理解。您可以使用以下预训练模型运行演示：

- **human-pose-estimation-0001**，一种人体姿态估计网络，可生成两个特征向量。该算法使用这两个特征向量预测人体姿态。

有关预训练模型的更多信息，请参阅[模型文档](#)。

输入帧高度按比例调整为模型高度，帧宽度也进行了相应的调整，以保留初始宽高比，并将其填充为 8 的倍数。

其他演示的目标为：

- 通过 OpenCV\* 将视频/摄像头用作输入
- 实现所有预测姿态的可视化

### 操作步骤

启动时，应用读取命令行参数并加载人体姿态估计模型。通过 OpenCV VideoCapture 获得帧后，应用执行人体姿态估计算法并显示结果。

**注：**默认情况下，Open Model Zoo 演示希望输入采用 BGR 通道顺序。如果经过训练的模型采用 RGB 顺序，您需要手动重新排列演示应用中的默认通道顺序，或者使用模型优化器工具（指定 `--reverse_input_channels` 参数）重新转换模型。有关参数的更多信息，请参阅“[何时反转输入通道顺序](#)”一节，[该节使用通用转换参数转换模型](#)。

### 运行

使用 `-h` 选项运行应用可生成下列使用信息：

```
1 ./human_pose_estimation_demo -h
2 .InferenceEngine:
3 . API version .....<version>
4 Build .....<number>
5
6 human_pose_estimation_demo [OPTION]
7 Options:
8
9 -h          Print a usage message.
10 -i "<path>" Required.Path to a video.Default value is "cam" to work with camera.
11 -m "<path>" Required.Path to the Human Pose Estimation model (.xml) file.
12 -d "<device>" Optional.Specify the target device for Human Pose Estimation (the list of
    available devices is shown below).Default value is CPU.Use "-d HETERO:<comma-
    separated_devices_list>" format to specify HETERO plugin.The application looks for a suitable
    plugin for the specified device.
13 -pc        Optional.Enable per-layer performance report.
14 -no_show   Optional.Do not show processed video.
15 -black     Optional.Show black background.
16 -r        Optional.Output inference results as raw values.
17 -u        Optional.List of monitors to show initially.
```

使用空选项列表运行应用程序可生成一条错误消息。

如欲运行演示，您可以使用公共或预训练模型。如欲下载预训练模型，请使用 OpenVINO [模型下载器](#) 或访问 <https://download.01.org/opencv/>。

注：使用经过训练的模型运行演示之前，请确保已使用 [模型优化器工具](#) 将模型转换为推理引擎格式 (\*.xml + \*.bin)。

例如，若要在 CPU 上执行推理，请运行以下命令：

```
1 ./human_pose_estimation_demo -i <path_to_video>/input_video.mp4 -m
   <path_to_model>/human-pose-estimation-0001.xml -d CPU
```

## 演示输出

该演示使用 OpenCV 显示包含预测姿态的帧和人体姿态估计演示的 **FPS** - 每秒帧数性能文本报告。

注：在 VPU 设备（英特尔® Movidius™ 神经计算棒、英特尔® 神经计算棒 2 和采用英特尔® Movidius™ VPU 的英特尔® 视觉加速器设计）上，该演示在以下可用的模型下载器拓扑上进行了测试：>\* human-pose-estimation-0001 其他模型可能在这些设备上产生意想不到的结果。

## 另请参阅

- [使用 Open Model Zoo 演示](#)
- [模型优化器](#)
- [模型下载器](#)

有关编译器优化的更多完整信息，请参阅我们的[优化声明](#)

## **支持**

[英特尔® OpenVINO™ 工具套件分发版的英特尔® 开发人员专区论坛](#)

## **Cookie**

[英特尔 Cookie 和类似技术声明](#)