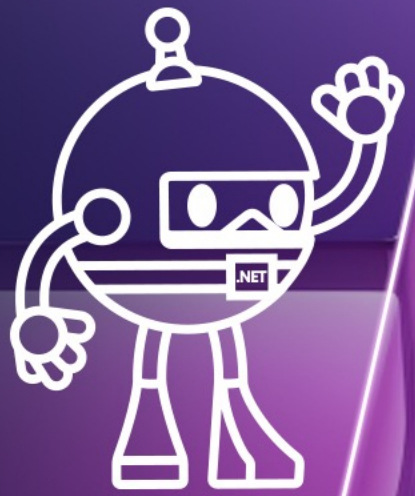
.NET Conf China 2023

2023/12/16
09:30 - 18:00

中国・北京



.NET中文社区

中国 · 北京

.NET Conf China 2023

玩转模型推理OpenVINO.NET

周杰 - 微软最有价值专家(MVP)

Microsoft® Most Valuable Professional

周杰 - 自我介绍

- .NET开发方向 微软最有价值专家(MVP)
- 2022年.NET Conf China主题《.NET玩转音视频操作FFmpeg》
- 2021年.NET Conf China主题《.NET玩转计算机视觉OpenCV》
- Github开源项目：https://github.com/sdcb
- 博客《.NET骚操作》 https://cnblogs.com/sdflysha
- B站 – 长沙周杰 https://space.bilibili.com/34943758332097
- QQ群
  - .NET骚操作交流群 495782587
  - C#/.NET计算机视觉技术交流 579060605

玩转模型推理OpenVINO.NET - 章节

- OpenVINO.NET项目背景与简介
- Demo效果演示
- PaddleOCR离线部署及跨平台部署

OpenVINO简介

- Open Visual Inference and Neural network Optimization
- 支持跨多种硬件平台，我亲测x64 CPU、GPU、ARM64都没问题
- 设计目的是提供一次性开发，到处部署的解决方案，强调开源和社区共享
- 内置图像预处理/后处理功能（PrePostProcessor），可简化推理流程
- 可直接加载多种模型，除自己的OpenVINO IR格式外，还支持直接加载：
  - Onnx（.onnx）
  - PaddlePaddle（.pdmodel）
  - 等等…
- 相比于其他解决方案， CPU性能方面有更好的表现（比如推理PaddleOCR方面）

.NET生态中的OpenVINO.NET

- 官方提供了Python、C++、C API，但没官方未提供.NET
- 相较于其他现有解决方案，OpenVINO.NET在下列领域展现出优势：
  - 全面性 – 为所有的242个C API提供了封装（使用CppSharp生成PInvoke代码）
  - 易用性 – 除了C API外，还提供了中高级API，用起来和Python/C++一样简单
  - 稳定性 – 提供完善的单元测试
  - 跨平台 – 支持Windows、Linux x64/ARM64，甚至安卓
  - 性能 – Intel原生开发，PaddleOCR推理速度比PaddleSharp快2倍（436ms -> 146ms）
  - 错误处理 – 所有C API的返回值都会做检查
- 提供示例丰富，原生提供简单易用的PaddleOCR模型推理代码
- 拥抱社区，可免费使用，基于Apache 2.0开源协议
- https://github.com/sdcb/OpenVINO.NET

OpenVINO.NET API设计原则

- **原汁原味的C API支持**：让喜爱底层控制和对性能敏感的开发者能直接使用OpenVINO的C API
- **高级API封装**：为所有底层C API提供简洁易用的C#高级封装，隐藏复杂性，加速开发
- **.NET社区命名规范**：确保中高级API符合.NET社区的命名和设计规范，让.NET开发者感到熟悉
- **异常和错误处理**：对C#高层API进行良好的异常和错误处理，提高代码的健壮性和可维护性
- **跨平台性**：支持在多种操作系统上使用，与OpenVINO的跨平台能力相匹配
- **性能优化**：利用C#特性（例如ReadOnlySpan<T>）进行性能优化，保证推理速度快速
- **文档与注释**：提供详尽的XML文档注释，帮助开发者理解每个API的用途和使用方法
- **单元测试覆盖**：编写全面的单元测试来确保代码质量和可靠性

安装OpenVINO.NET

- 最简单的方法是使用NuGet包
- （而非下载源代码——但欢迎了解细节和star🌟）
- 安装步骤：
  1. 安装核心包：Sdcb.OpenVINO
  2. 安装平均动态库包，比如：
     - Windows – Sdcb.OpenVINO.runtime.win-x64
     - Ubuntu 22.04 x64 - Sdcb.OpenVINO.runtime.ubuntu.22.04-x64
- 观看使用示例和教程：
  - 4个示例：https://github.com/sdcb/OpenVINO.NET
  - 视频教程：https://space.bilibili.com/3494375833209736

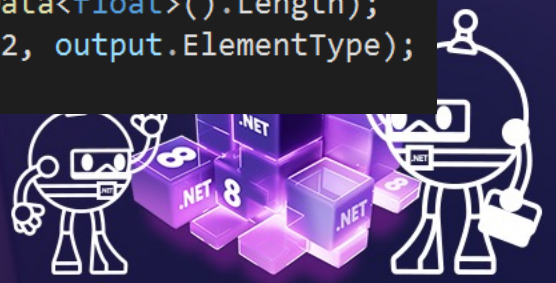
易用性展示

OpenVINO.NET C API

```
[Fact]
0 个引用|sdcb，69 天前|1 名作者，1 项更改
public unsafe void PreprocessSteps()
{
    ov_core* core = null;
    ov_model* model = null;
    ov_output_const_port* outputPort = null;
    ov_output_const_port* inputPort = null;
    ov_shape_t shape = default;
    ov_tensor* tensor = null;
    ov_preprocess_prepostprocessor* preprocessor = null;
    ov_preprocess_input_info* inputInfo = null;
    ov_preprocess_input_tensor_info* inputTensorInfo = null;
    ov_layout* inputLayout = null;
    ov_preprocess_preprocess_steps* preprocessSteps = null;
    ov_preprocess_input_model_info* modelInfo;
    ov_layout* modelLayout = null;
    ov_model* newModel = null;
    ov_compiled_model* compiledModel = null;
    ov_infer_request* inferRequest = null;
    ov_tensor* outputTensor = null;
    ov_preprocess_output_info* outputInfo = null;
    ov_preprocess_output_tensor_info* outputTensorInfo = null;

    try
    {
        Check(ov_core_create(&core));
        fixed (byte* modelPathPtr = Encoding.UTF8.GetBytes(_modelFile))
        {
            Check(ov_core_read_model(core, modelPathPtr, null, &model));
            Check(ov_model_const_output(model, &outputPort));
            Check(ov_model_const_input(model, &inputPort));
            using Mat mat = Cv2.ImRead(@"./assets/text.png");
            Cv2.CopyMakeBorder(mat, mat, 0, 960 - mat.Rows, 0, 960 - mat.Cols, BorderTypes.Constant, Scalar.Black);
            mat.ConvertTo(mat, MatType.CV_32FC3, 1.0 / 255);
            //using Mat normalized = Normalize(mat);
            //float[] floatData = ExtractMat(normalized);
            long* dims = stackalloc long[4] { 1, mat.Height, mat.Width, 3 };
            Check(ov_shape_create(4, dims, &shape));
            Check(ov_tensor_create_from_host_ptr(ov_element_type_e.F32, shape, (void*)mat.Data, &tensor));
            Check(ov_preprocess_prepostprocessor_create(model, &preprocessor));
            Check(ov_preprocess_prepostprocessor_get_input_info_by_index(preprocessor, 0, &inputInfo));

            Check(ov_preprocess_input_info_get_tensor_info(inputInfo, &inputTensorInfo));
            Check(ov_preprocess_input_tensor_info_set_from(inputTensorInfo, tensor));
            byte* inputLayoutDesc = stackalloc byte[4] { (byte)'N', (byte)'H', (byte)'W', (byte)'C' };
            Check(ov_layout_create(inputLayoutDesc, &inputLayout));
            Check(ov_preprocess_input_tensor_info_set_layout(inputTensorInfo, inputLayout));

            Check(ov_preprocess_input_info_get_preprocess_steps(inputInfo, &preprocessSteps));

            //Check(ov_preprocess_preprocess_steps_resize(preprocessSteps, ov_preprocess_resize_algorithm_e.RESIZE_LINEAR));
            Check(ov_preprocess_input_info_get_model_info(inputInfo, &modelInfo));

            byte* modelLayoutDesc = stackalloc byte[4] { (byte)'N', (byte)'C', (byte)'H', (byte)'W' };
            Check(ov_layout_create(modelLayoutDesc, &modelLayout));
            Check(ov_preprocess_input_model_info_set_layout(modelInfo, modelLayout));

            Check(ov_preprocess_prepostprocessor_get_output_info_by_index(preprocessor, 0, &outputInfo));
            Check(ov_preprocess_output_info_get_tensor_info(outputInfo, &outputTensorInfo));
            Check(ov_preprocess_output_set_element_type(outputTensorInfo, ov_element_type_e.F32));

            Check(ov_preprocess_prepostprocessor_build(preprocessor, &newModel));

            fixed (byte* deviceName = Encoding.UTF8.GetBytes("CPU"))
            {
                Check(ov_core_compile_model(core, newModel, deviceName, 0, &compiledModel));
            }

            Check(ov_compiled_model_create_infer_request(compiledModel, &inferRequest));
            Check(ov_infer_request_set_input_tensor_by_index(inferRequest, 0, tensor));
            Check(ov_infer_request_infer(inferRequest));
            Check(ov_infer_request_get_output_tensor_by_index(inferRequest, 0, &outputTensor));
            void* data;
            Check(ov_tensor_data(outputTensor, &data));
            nint dataSize;
            Check(ov_tensor_get_byte_size(outputTensor, &dataSize));
            using Mat result = new(960, 960, MatType.CV_32FC1, (IntPtr)data);
            result.ConvertTo(result, MatType.CV_8SC1, 255);
        }
    }
    finally
    {
        if (outputTensorInfo != null) ov_preprocess_output_tensor_info_free(outputTensorInfo);
        if (outputInfo != null) ov_preprocess_output_info_free(outputInfo);
        if (modelLayout != null) ov_layout_free(modelLayout);
        if (inputTensorInfo != null) ov_preprocess_input_tensor_info_free(inputTensorInfo);
        if (inputInfo != null) ov_preprocess_input_info_free(inputInfo);
        if (preprocessor != null) ov_preprocess_prepostprocessor_free(preprocessor);
        if (tensor != null) ov_tensor_free(tensor);
        if (shape.dims != null) ov_shape_free(&shape);
        if (outputPort != null) ov_output_const_port_free(outputPort);
        if (inputPort != null) ov_output_const_port_free(inputPort);
        if (model != null) ov_model_free(model);
        if (core != null) ov_core_free(core);
    }
}
```

OpenVINO.NET 中高级API

```
[Fact]
0 个引用|sdcb，69 天前|1 名作者，1 项更改
public void MinInfer()
{
    using OVCore c = new();
    using CompiledModel cm = c.CompileModel(_modelFile);
    using InferRequest r = cm.CreateInferRequest();
    using Tensor input = r.Inputs.Primary;
    input.Shape = new Shape(1, 3, 32, 64);

    r.Run();

    using Tensor output = r.Outputs.Primary;
    Assert.Equal(new Shape(1, 1, 32, 64), output.Shape);
    Assert.Equal(32 * 64, output.GetData<float>().Length);
    Assert.Equal(ov_element_type_e.F32, output.ElementType);
}
```



NuGet包简介

Core packages

Package	Version	Description
Sdcb.OpenVINO	nuget v0.6.1	.NET PInvoke interface
Sdcb.OpenVINO.Extensions.OpenCvSharp4	nuget v0.6.1	OpenVINO OpenCvSharp4 extensions

Platform shared runtime packages

Package	Version	Description
Sdcb.OpenVINO.runtime.centos.7-x64	nuget v2023.2.0	Runtime for CentOS 7 x64
Sdcb.OpenVINO.runtime.linux-arm	nuget v2023.2.0	Runtime for Debian 9+ ARM
Sdcb.OpenVINO.runtime.linux-arm64	nuget v2023.2.0	Runtime for Debian 9+ ARM64
Sdcb.OpenVINO.runtime.rhel.8-x64	nuget v2023.2.0	Runtime for RHEL 8 x64
Sdcb.OpenVINO.runtime.ubuntu.18.04-x64	nuget v2023.2.0	Runtime for Ubuntu 18.04 x64
Sdcb.OpenVINO.runtime.ubuntu.20.04-x64	nuget v2023.2.0	Runtime for Ubuntu 20.04 x64
Sdcb.OpenVINO.runtime.ubuntu.22.04-x64	nuget v2023.2.0	Runtime for Ubuntu 22.04 x64
Sdcb.OpenVINO.runtime.android-arm64	nuget v2023.1.0	Runtime for Android ARM64
Sdcb.OpenVINO.runtime.win-x64	nuget v2023.2.0	Runtime for Windows x64

https://www.nuget.org/packages?q=Sdcb.OpenVINO

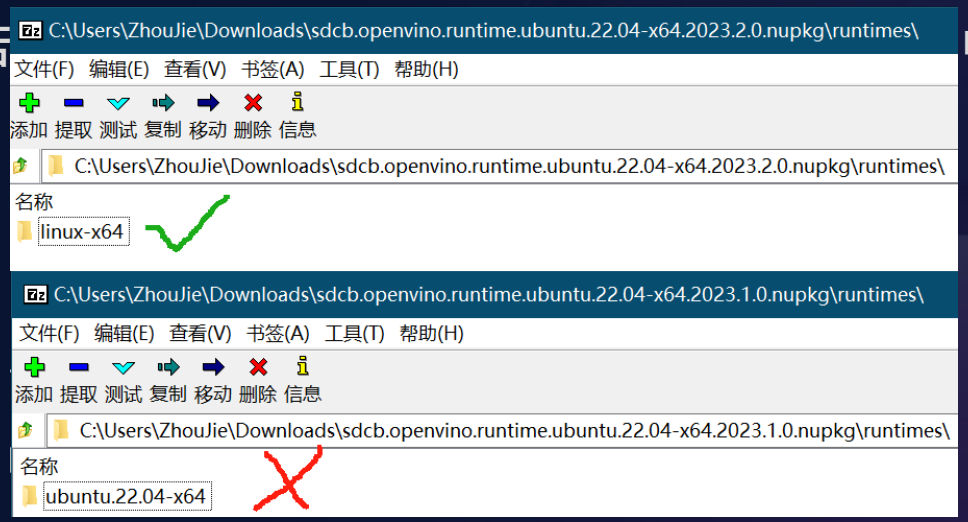
OpenCvSharp4 mini runtime

Id	Version	Size	OS	Arch
Sdcb.OpenCvSharp4.mini.runtime.centos.7-arm64	nuget v4.8.0.20231125	3.23MB	CentOS 7	ARM64
Sdcb.OpenCvSharp4.mini.runtime.centos.7-x64	nuget v4.8.0.20231125	16.75MB	CentOS 7	x64
Sdcb.OpenCvSharp4.mini.runtime.debian.11-arm64	nuget v4.8.0.20231125	4.05MB	Debian 11	ARM64
Sdcb.OpenCvSharp4.mini.runtime.debian.11-x64	nuget v4.8.0.20231125	18.13MB	Debian 11	x64
Sdcb.OpenCvSharp4.mini.runtime.debian.12-arm64	nuget v4.8.0.20231125	4.18MB	Debian 12	ARM64
Sdcb.OpenCvSharp4.mini.runtime.debian.12-x64	nuget v4.8.0.20231125	17.47MB	Debian 12	x64
Sdcb.OpenCvSharp4.mini.runtime.ubuntu.22.04-arm64	nuget v4.8.0.20231125	4.18MB	Ubuntu 22.04	ARM64
Sdcb.OpenCvSharp4.mini.runtime.ubuntu.22.04-x64	nuget v4.8.0.20231125	17.47MB	Ubuntu 22.04	x64
Sdcb.OpenCvSharp4.mini.runtime.android-arm64	nuget v4.8.0.20230708	4.04MB	Android	ARM64
Sdcb.OpenCvSharp4.mini.runtime.android-x64	nuget v4.8.0.20230708	5.9MB	Android	x64

https://www.nuget.org/packages?q=Sdcb.OpenCvSharp4

.NET 8升级避坑

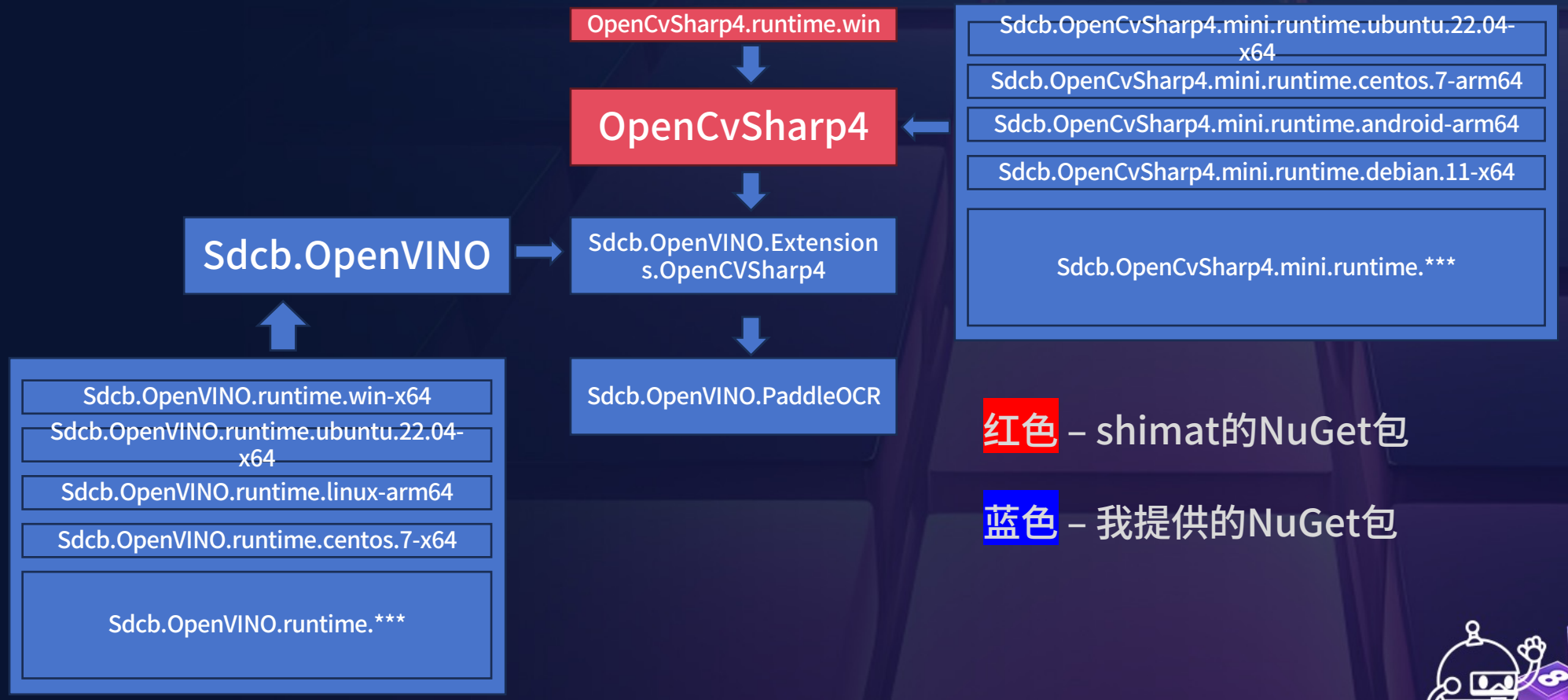
- https://learn.microsoft.com/zh-cn/dotnet/core/compatibility/deployment/8.0/rid-asset-list
- NuGet包中的Runtime identifier不再支持具体的操作系统
- 而是统一为win/linux/mac等，而非win10/ubuntu22.04/mac-11
- NuGet包的作者应该遵守这一新规则（如下图）
- 若 中



XML 在csproj项目中，加入：

```xml
<ItemGroup>
  <RuntimeHostConfigurationOption Include="System.Runtime.Loader.UseRidGraph" Value="true" />
</ItemGroup>
```

项目架构

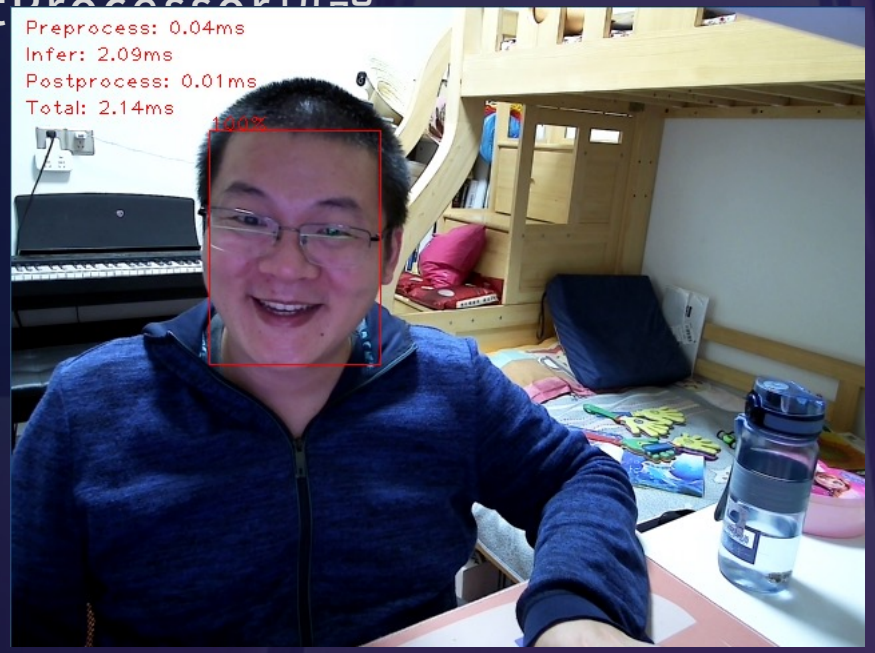


红色 – shimat的NuGet包

蓝色 – 我提供的NuGet包

OpenVINO.NET示例一：人脸检测

- 完整源代码：https://github.com/sdcb/mini-openvino-facedetection
- 使用的是OpenVINO - IR模型
- Demo展示了OpenVINO的PrePostProcessor功能
- 依赖于OpenCvSharp4读取摄像头



OpenVINO.NET示例二：物体检测

- 完整源代码：https://github.com/sdcb/sdcb-openvino-yolov8-det
- 使用的是yolov8-det转onnx模型

OpenVINO.NET示例三：物体分类

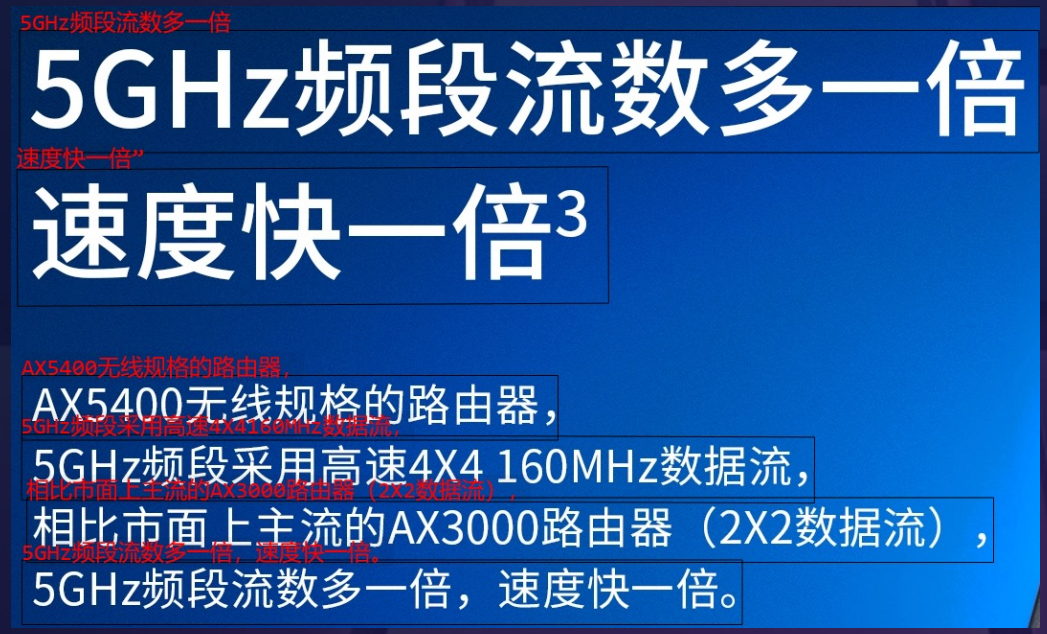
- 完整源代码：https://github.com/sdcb/sdcb-openvino-yolov8-cls
- 使用的是yolov8-cls转onnx模型

输出如下：

```
class id=hen, score=0.59
preprocess time: 0.00ms
infer time: 1.65ms
postprocess time: 0.49ms
Total time: 2.14ms
```

OpenVINO.NET示例四：PaddleOCR

- 主仓库提供了便利化代码：https://github.com/sdcb/OpenVINO.NET/tree/master/projects/PaddleOCR
- 代码示例：https://github.com/sdcb/mini-openvino-paddleocr
- 使用了3个PaddlePaddle（.pdmodel）模型混合进行推理
- 支持x64 CPU/GPU/ARM 64
- 支持离线部署
- 支持跨平台



PaddleOCR入门及离线部署 - 视频教程

内容简介：

https://www.bilibili.com/video/BV1bM411f74Z/

- 从C# HelloWorld到OCR Demo
- OCR模型下载和离线运行设置
- 如何获取OCR方框位置信息
- 性能调参实践
- 如何GPU运行并演示
- 按固定形状输入编译（“静态图”）优化性能

我最近一年的其它开源项目（欢迎star🌟）

项目	链接
Sdcb.Arithmetic 高精度数值计算	https://github.com/sdcb/Sdcb.Arithmetic
PaddleSharp	https://github.com/sdcb/PaddleSharp
Sdcb.FFmpeg	https://github.com/sdcb/Sdcb.FFmpeg
Sdcb.LibRaw	https://github.com/sdcb/Sdcb.LibRaw
Sdcb.SparkDesk 讯飞星火非官方.NET SDK	https://github.com/sdcb/Sdcb.SparkDesk
Sdcb.WenXinQianFan 文心千帆非官方.NET SDK	https://github.com/sdcb/Sdcb.WenXinQianFan
Sdcb.StabilityAI Stability AI非官方.NET SDK	https://github.com/sdcb/Sdcb.StabilityAI

PaddleOCR跨平台部署 - 视频教程

覆盖的操作系统或平台：

- Ubuntu 22.04 x64
- Debian 12 x64
- Debian 11 x64
- CentOS 7 x64
- Ubuntu 22.04 ARM64
- Debian 11 ARM64
- 香橙派4-LTS（基于国产ARM64-RK3399芯片）

https://www.bilibili.com/video/BV1R64y1L7Sv/

让.NET成为第一等公民

（任何好玩的领域）

谢谢大家

答疑Q/A

QQ群：495782587

周杰 - 微软最有价值专家(MVP)

PPT下载：https://io.starworks.cc:88/cv-public/2023/2023-zhoujie-openvino.pdf

Microsoft® MVP Most Valuable Professional