Class SegmentationPostprocessorOp

Table of contents

- Defined in File segmentation_postprocessor.hpp

Inheritance Relationships

Base Type

- `public holoscan::Operator` (Class Operator)

Class Documentation

class SegmentationPostprocessorOp : public holoscan::Operator

Operator carrying out post-processing operations on segmentation outputs.

==Named Inputs==

- **in_tensor** : `nvidia::gxf::Tensor`
  - Expects a message containing a 32-bit floating point device tensor with name `in_tensor_name`. The expected data layout of this tensor is HWC, NCHW or NHWC format as specified via `data_format`. If batch dimension, N, is present it should be size 1.

==Named Outputs==

- **out_tensor** : `nvidia::gxf::Tensor`
  - Emits a message containing a device tensor named “out_tensor” that contains the segmentation labels. This tensor will have unsigned 8-bit integer data type and shape (H, W, 1).

==Parameters==

- **allocator**: Memory allocator to use for the output.
- **in_tensor_name**: Name of the input tensor. Optional (default: `""`).
- **network_output_type**: Network output type (e.g. ‘softmax’). Optional (default: `"softmax"`).

- **data_format**: Data format of network output. Optional (default: `"hwc"`).
- **cuda_stream_pool**: `holoscan::CudaStreamPool` instance to allocate CUDA streams. Optional (default: `nullptr`).

==Device Memory Requirements==

When used with a `BlockMemoryPool`, this operator requires only a single device memory block (`storage_type` = 1) of size `height * width` bytes.

Public Functions

HOLOSCAN_OPERATOR_FORWARD_ARGS (SegmentationPostprocessorOp)
SegmentationPostprocessorOp()=default

virtual void setup(OperatorSpec &spec) override

Define the operator specification.

Parameters

spec – The reference to the operator specification.

virtual void start() override

Implement the startup logic of the operator.

This method is called multiple times over the lifecycle of the operator according to the order defined in the lifecycle, and used for heavy initialization tasks such as allocating memory resources.

virtual void compute(InputContext &op_input, OutputContext &op_output, ExecutionContext &context) override

Implement the compute method.

This method is called by the runtime multiple times. The runtime calls this method until the operator is stopped.

Parameters

- **op_input** – The input context of the operator.

- **op_output** – The output context of the operator.
- **context** – The execution context of the operator.