Program Listing for File segmentation_postprocessor.hpp

Return to documentation for file (
include/holoscan/operators/segmentation_postprocessor/segmentation_postprocessor.h
)

/* * SPDX-FileCopyrightText: Copyright (c) 2022-2024 NVIDIA CORPORATION & AFFILIATES. All rights reserved. * SPDX-License-Identifier: Apache-2.0 * * Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ #ifndef HOLOSCAN_OPERATORS_SEGMENTATION_POSTPROCESSOR_POSTPROCESSOR_HPP #define HOLOSCAN_OPERATORS_SEGMENTATION_POSTPROCESSOR_POSTPROCESSOR_HPP #include <memory> #include <string> #include <utility> #include "holoscan/core/io_context.hpp" #include "holoscan/core/io_spec.hpp" #include "holoscan/core/operator.hpp" #include "holoscan/core/operator_spec.hpp" #include "holoscan/utils/cuda_stream_handler.hpp" #include "segmentation_postprocessor.cuh" using holoscan::ops::segmentation_postprocessor::DataFormat; using holoscan::ops::segmentation_postprocessor::NetworkOutputType; namespace holoscan::ops { class SegmentationPostprocessorOp : public Operator { public: HOLOSCAN_OPERATOR_FORWARD_ARGS(SegmentationPostprocessorOp) SegmentationPostprocessorOp() = default; *// TODO(gbae): use std::expected* void setup(OperatorSpec& spec) override; void start() override; void compute(InputContext& op_input, OutputContext& op_output, ExecutionContext& context) override; private: NetworkOutputType network_output_type_value_; DataFormat data_format_value_; Parameter<holoscan::IOSpec*> in_; Parameter<holoscan::IOSpec*> out_; Parameter<std::shared_ptr<Allocator>> allocator_; Parameter<std::string> in_tensor_name_; Parameter<std::string> network_output_type_; Parameter<std::string> data_format_; CudaStreamHandler cuda_stream_handler_; }; } *// namespace holoscan::ops* #endif/*

```
HOLOSCAN_OPERATORS_SEGMENTATION_POSTPROCESSOR_POSTPROCESSOR_HPP
*/
```