



Program Listing for File infer_manager.hpp

[Return to documentation for file \(modules/holoinfer/src/manager/infer_manager.hpp\)](#)

```
/* * 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_INFER_MANAGER_H #define _HOLOSCAN_INFER_MANAGER_H
#include <functional> #include <future> #include <iostream> #include <map>
#include <memory> #include <set> #include <string> #include <holoinfer.hpp>
#include <holoinfer_buffer.hpp> #include <holoinfer_constants.hpp> #include
<holoinfer_utils.hpp> #include <infer/infer.hpp> #if use_onnxruntime #include
<infer/onnx/core.hpp> #endif #if use_torch #include <infer/torch/core.hpp> #endif
#include <infer/trt/core.hpp> #include <params/infer_param.hpp> namespace
holoscan { namespace inference { class ManagerInfer { public: ManagerInfer();
~ManagerInfer(); InferStatus
set_inference_params(std::shared_ptr<InferenceSpecs>& inference_specs);
InferStatus execute_inference(DataMap& preprocess_data_map, DataMap&
output_data_map); InferStatus run_core_inference(const std::string& model_name,
DataMap& permodel_preprocess_data, DataMap& permodel_output_data); void
cleanup(); DimType get_input_dimensions() const; DimType get_output_dimensions()
const; private: bool parallel_processing_ = false; bool cuda_buffer_in_ = false; bool
cuda_buffer_out_ = false; bool mgpu_p2p_transfer = true; std::map<std::string,
std::map<std::string, cudaStream_t>> input_streams_gpudt; std::map<std::string,
std::map<std::string, cudaStream_t>> output_streams_gpudt; std::map<std::string,
std::map<std::string, cudaStream_t>> input_streams_device; std::map<std::string,
std::map<std::string, cudaStream_t>> output_streams_device; std::map<std::string,
std::unique_ptr<Params>> infer_param_; std::map<std::string,
std::unique_ptr<InferBase>> holo_infer_context_; DimType models_input_dims_;
std::map<std::string, DataMap> mgpu_output_buffer_; std::map<std::string,
DataMap> mgpu_input_buffer_; unsigned int frame_counter_ = 0; int device_gpu_dt
```

```
= 0; DimType models_output_dims_; inline static std::map<std::string,
holoinfer_backend> supported_backend_{ {"onnxrt", holoinfer_backend::h_onnx},
{"trt", holoinfer_backend::h_trt}, {"torch", holoinfer_backend::h_torch}}; };
std::shared_ptr<ManagerInfer> g_manager; std::map<std::string,
std::shared_ptr<ManagerInfer>> g_managers; } // namespace inference } //
namespace holoscan #endif
```

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024