Program Listing for File core.hpp

Return to documentation for file (modules/holoinfer/src/infer/trt/core.hpp)

/* * SPDX-FileCopyrightText: Copyright (c) 2022-2023 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. */ #include <cstring> #include <iostream> #include <memory> #include <string> #include <vector> #include <NvInfer.h> #include <holoinfer_constants.hpp> #include <infer/infer.hpp> #include "utils.hpp" namespace holoscan { namespace inference { class TrtInfer : public InferBase { public: TrtInfer(const std::string& model_path, const std::string& model_name, int device_id, bool enable_fp16, bool is_engine_path, bool cuda_buf_in, bool cuda_buf_out); ~TrtInfer(); InferStatus do_inference(const std::vector<std::shared_ptr<DataBuffer>>& input_data, std::vector<std::shared_ptr<DataBuffer>>& output_buffer); std::vector<std::vector<int64_t>> get_input_dims() const; std::vector<std::vector<int64_t>> get_output_dims() const; std::vector<holoinfer_datatype> get_input_datatype() const; std::vector<holoinfer_datatype> get_output_datatype() const; void cleanup() {} private: std::string model_path_; std::string model_name_; std::vector<std::vector<int64_t>> input_dims_; std::vector<std::vector<int64_t>> output_dims_; std::vector<holoinfer_datatype> in_data_types_; std::vector<holoinfer_datatype> out_data_types_; bool enable_fp16_; bool cuda_buf_in_; bool cuda_buf_out_; bool is_engine_path_; bool initialize_parameters(); bool load_engine(); std::unique_ptr<nvinfer1::ICudaEngine> engine_ = nullptr; std::unique_ptr<nvinfer1::IExecutionContext> context_ = nullptr; NetworkOptions network_options_; int device_id_; Logger logger_; std::shared_ptr<DataBuffer> input_buffer_; std::shared_ptr<DataBuffer> output_buffer_; std::string engine_path_; cudaStream_t cuda_stream_ = nullptr; std::unique_ptr<nvinfer1::IRuntime> infer_runtime_; }; } *// namespace inference* } *// namespace holoscan*