



Author: NeuroCheck GmbH – support@neurocheck.com

Content: This document contains important notes and usage information for NeuroCheck's Neural Networks plug-in PI_NeuralNetworkTools.

Notes: This document is not part of the official product documentation of the NeuroCheck software.

NeuroCheck GmbH does not accept responsibility for the correctness, accuracy or completeness of the information provided in this document.

Windows is a registered trademark of Microsoft corporation.

CUDA and cuDNN are registered trademarks of NVIDIA corporation.

Table of contents:

1.	Introduction	3
2.	Further Information	3
3.	Versions	3
4.	Installation	4
5.	Usage Notes	5
5.1.	Neural Network Design	5
5.1.1.	Model Format.....	5
5.1.2.	Input Layer	5
5.1.3.	Output Layer:	5
5.2.	Check Functions	6
6.	Support and Contact	7

1. Introduction

NeuroCheck's PI_NeuralNetworkTools plug-in enables you to leverage the full power and possibilities of Artificial Intelligence (AI) in image processing. It seamlessly integrates Neural Networks into your visual inspection task and further extends NeuroCheck as a platform for hardware, visualization and process integration.

What this plug-in is:

It is a runtime for pretrained models in ONNX¹ (*.onnx) format and supports the following network types:

- Classification
- Object Detection
- Semantic Segmentation
- Anomaly Detection
- Auto Encoder

What this plug-in is NOT:

The plug-in does not offer any labelling, design, training or evaluation pipeline for creating Neural Networks. You may use your preferred tools for these tasks, for example TensorFlow, PyTorch or Keras or others. For a complete list of AI frameworks supporting ONNX please refer to the ONNX website².

2. Further Information

This whitepaper only contains some basic information about PI_NeuralNetworkTools. Please refer to the following resources for further details:

- PI_NeuralNetworkTools help file: delivered with the plug-in
- Application examples: delivered with the plug-in
- NeuroCheck Software Support: support@neurocheck.com

3. Versions

PI_NeuralNetworkTools uses ONNX 1.9.0 and is available in two different versions:

- **CPU** for inference on CPU and
- **GPU** for inference using a graphics card as a hardware accelerator

The plug-in supports NeuroCheck 6.2.

Which version you need strongly depends on your application. The CPU version does not require extra hardware (a CPU with AVX support recommended though). The inference speed, however, is slower compared to the GPU version. The GPU version requires an NVIDIA Graphics Card and is recommended for high-speed applications.

If you are not sure which version you should use for your very application, do not hesitate to get in contact with us.

¹ Open Neural Network Exchange

² <https://onnx.ai/supported-tools.html>

4. Installation

The following section provides a short description of the installation process for both versions. Please refer to the help file of PI_NeuralNetworkTools for further details.

Prerequisites CPU & GPU:

- Onnx Runtime: Windows 10 and Windows 7
- Microsoft Visual C++ Redistributable 2019

CPU Version:

Just copy the contents of the ZIP-file PI_NeuralNetworkTools.NET.CPU.zip to the Software Extensions\PlugIns directory of your NeuroCheck project. You might have to install Microsoft Visual C++ Redistributable 2019, which is included in the ZIP file (vc_redist.x64.exe). Depending on your operating system, also copy the contents of the folder Prerequisites\Win7 or \Win10 to your plug-in directory.

GPU Version:

Copy the contents of the ZIP-file PI_NeuralNetworkTools.NET.GPU.zip to the Software Extensions\PlugIns directory of your NeuroCheck project. Depending on your operating system, also copy the Onnx Runtime from the folder Prerequisites\Win7 or \Win10 to your plug-in directory.

The GPU version of the plug-in requires NVIDIA CUDA drivers and NVIDIA cuDNN framework. So please make sure you install the following packages:

- NVIDIA CUDA 11.6.0
- NVIDIA CuDNN 8.4.1.50

There are installers for CUDA and CuDNN in the Prerequisites folder included in the ZIP file. Please select according to your operating system. The plug-in supports NVIDIA GPUs with a **Compute Capability of 5.0 or higher**. Please refer to [nvidia.com³](https://developer.nvidia.com/cuda-gpus) for a complete list.

³ <https://developer.nvidia.com/cuda-gpus>

5. Usage Notes

5.1. Neural Network Design

For a detailed description of the design requirements please refer to the plug-in help file. The following section contains some hints you should consider for design and training.

5.1.1. Model Format

PI_NeuralNetworkTools supports ONNX models. Please consult the documentation of your preferred AI framework on how to export models in this format.

5.1.2. Input Layer

The images are passed to the runtime in 4-dimensional **NHWC and NCHW** format, where N is the number of batches, H is the image height, W corresponds to the image width and C is the number of channels (e.g. 1 for gray scale, 3 for color), so please design your input layer accordingly. Please also not that **NCHW** format results in much better model performance due to optimized memory layout.

The input layer should be of type Uint8 or Uint16. Other formats are supported, but require a cast to the target format. This cast will have a huge impact on inference performance.

5.1.3. Output Layer:

The output layer design depends on the network type (Classification, Object Detection etc.). Please refer to the plug-in help file for more detailed information.

Network Type	Output Layer Hints
Classification	Float32 bit array of class scores
Object Detection	Number of detections as Int32 or Float32, bounding box coordinates as Float32 array, scores as Float32 array, classes as Float32 array
Auto Encoder	Image as Uint8 array
Segmentation	Uint8 array with the dimensions n x h x w (n = number of batches, h = height of the segmentation map, w = width of the segmentation map)
Anomaly Detection	Float32 array of anomaly scores and a NHWC anomaly map array in UINt8 format.

5.2. Check Functions

PI_NeuralNetworkTools implements check functions for five different neural network types:

- Classify Objects
- Detect Objects
- Segment Objects
- Detect Anomalies
- Auto Encode Objects

All check functions expect an image and a ROI list as input objects. The model evaluates each ROI item in the list. If the input ROI list contains more than one item, it is possible to execute the inference in parallel. Adjust the *Batch Size* in the check function's parameter dialog accordingly. Please make sure to test the batch size you configured before production. It might occur that the GPU's memory is not large enough to handle the batch size. You can check your batch size configuration by hitting the *Test-Button* in the parameter dialog.

Classify Objects:

Use this check function for classification tasks. The check function assigns resulting classes to the input ROIs.

Detect Objects:

For object detection tasks. It expects an image and a ROI list as input objects. The check function creates rectangular ROIs for each detected object and generates a ROI list as output.

Segment Objects:

Performs semantic segmentation on the input ROIs and the underlying image. The check function creates boundary ROIs for each detected object and generates a ROI list as output.

Detect Anomalies:

For anomaly detection models. Assigns an Anomaly Score to each input ROI. The Anomaly Map output image may be used for visualization of anomalies; it is **not** possible, however, to quantify the degree of anomaly processing this image.

6. Support and Contact

Please do not hesitate to contact us in case of any further questions. We are looking forward to your feedback.

Phone: +49 (0) 7146 - 89 56-40

E-Mail: support@neurocheck.com

Web: www.neurocheck.com