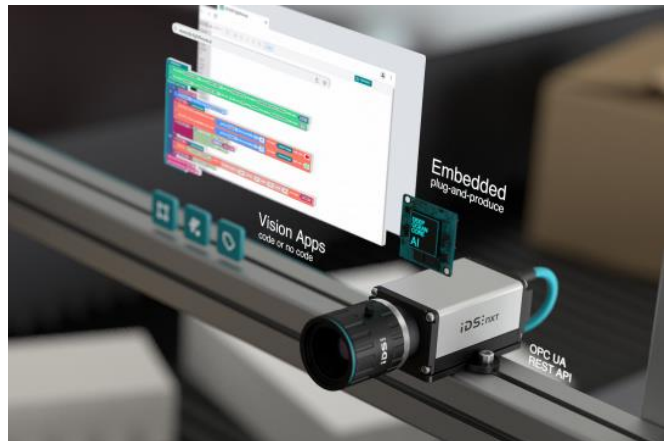


IDS NXT rio (and IDS lighthouse)

This document gives a brief overview of the IDS NEXT. It is based on evaluation of the sensor performed by Stiftelsen Adopticum in a project financed by Kempestiftelsen. For more information about the sensor, please feel free to contact Adopticum.

The IDS NXT rio is a vision camera, suitable for industrial use. It is programmable with additional software meant to streamline the process of creating Convolutional neural networks (CNNs) for the camera to use. The cloud-based vision studio *IDS lighthouse* provides the necessary functionality for uploading images, labelling images, training and creating a CNN, guiding the user through all necessary steps in a clear workflow. The camera aims to provide the means to solve problems using Artificial Intelligence for vision applications (object detection, classification, etc.), even for users with limited experience and knowledge of the specifics of Deep learning, or even coding.



The IDS NXT rio camera can be configured using the software IDS NXT cockpit. With this software, it is easy to get the camera connected and running, adjusting the settings and uploading available vision applications (.vapp files) to the camera for performing tasks and returning a result as output. IDS provides already trained CNNs for analysing images, included in vision applications for the user to try. Examples:

- Classifier: Identifies many different objects in the camera images and returns the detected object class name and probability of the detection (see Figure 1 below).
- Object-detector: Detecting objects and their positions within the camera images. (see PersonDetector in Figure 2 below).

The user can setup an FTP client to stream images from the camera. These two applications described above have settings for choosing what images are sent, like using a probability threshold for the detections in the images as a condition.

Also available is the functionality to use an OPC UA server integrated into the camera which allows clients to communicate with the camera and get the results from its vision application.

With IDS lighthouse all the steps necessary for the training of an CNN are included and the user is guided step by step, making the processes simple and without needing coding experience. However, knowledge of what is needed to get a well-trained CNN is still something that is needed to get a well performing AI model, even though IDS lighthouse does provide important information regarding the

acquirements for the input data for a CNN creation to be possible. There are 3 different project types:

- Learn: Very limited functionality, where sample projects show the general steps needed for creating a convolutional neural network with predefined data sets.
- Configure: Take predefined user cases and create your project from them. You can use your own data sets to train a CNN for detecting the objects of your choice.
- Create_new: Allows you to create your own CNNs with the data input of your choice for your use case. You can use the block-based editor, where you can add the functionality of your choice to create your own vision apps that can use your CNN and then be uploaded to the IDS NXT camera. This functionality could include setting an output high or low depending on the detection results or add a probability limit below which the detected objects are ignored.

All these project types take the user through the workflow step by step all the way for the training of a CNN. Configure and Create_new-project types can also create complete vision applications that can then be uploaded to the IDS NXT camera. You can also convert the projects with limited functionality to Create_new-project types. By converting example Configure-projects you can get a full application ready in the block-based editor, which you can analyse to get an idea of how the block programming can be done. There are tutorial videos and also documentation available for download on IDS webpage (you need to create a free user account): <https://en.ids-imaging.com/download-details/AS00029.html>

Existing data sets with prepared/labelled images using the format *.lhd* can also be imported to IDS lighthouse. If the user has created one data set that set can easily be imported into future projects as well.

Using IDS lighthouse, a simple power strip analyser application was created. If the top area (with the cable connectors) can be seen the CNN will return the result OK, but if the top area cannot be seen in the camera image the result will be NOK (see Figure 3 below). An example project OK or NOK was used as a basis for the project. A vision application was created and uploaded to the camera which performed well (for this very simple application).

An IDX NXT Experience Kit, for the price of about 2 100 Euro, will get the user up and running quickly, ready to learn how to create their own CNNs. The kit includes all the hardware needed to get started with the camera, together with a 6-month license for IDS lighthouse (one free month trial is available first). After that the license fee needs to be paid for continuing to use IDS lighthouse to train more CNNs.

While experience and knowledge of the important factors of taking images (and the importance of the background scene and variation of the same), labelling data and what to even choose to detect and use it for will still be important and a learning process for beginners, the steps needed using software to label images, training a CNN and using it to create a vision application are all collected in a simple workflow with instructions in IDS lighthouse. And together with an industrial IDS NXT camera, these applications can be deployed as a vision system tailor made for your needs, even without programming knowledge. But the resulting CNN performance does depend on the training (collecting enough images where the objects to detect are included, accurate labelling, etc.) and it is important to remember that the images the camera can capture from its intended position at the

site where the CNN is to be used, need to allow for the CNN to do its job (clear view of the objects to detect, enough ambient light, temperature protection preventing overheating or freezing, etc.). Images from the intended use case site on actual objects the user wants to detect should, for example, be included in the data set used for training.



Figure 1: Images and CNN output from an IDS NXT rio camera. The camera image with results displayed in the bottom of and below the image. As seen the neural network has several different categories/labels that it gets a match for, but with different probabilities. The highest probability is for ballpoint pen, which is shown as the top result, and is the correct one in this case. While the prediction probability is not constant between camera frames, and can differ 10 % or more for this CNN, the big difference seen between the top and bottom images include (slightly) different camera angles and whether the tip of the pen is withdrawn or not. The bottom image, with the tip extracted, shows a more stable result with around 81 % probability for the ballpoint pen category, while the top image shows 36 %, with the category syringe being close with 30 % probability. This illustrates how an AI model can work and perform. The performance does depend on the training and the data used for the training, but do not expect 100 % detection probability for an AI model.

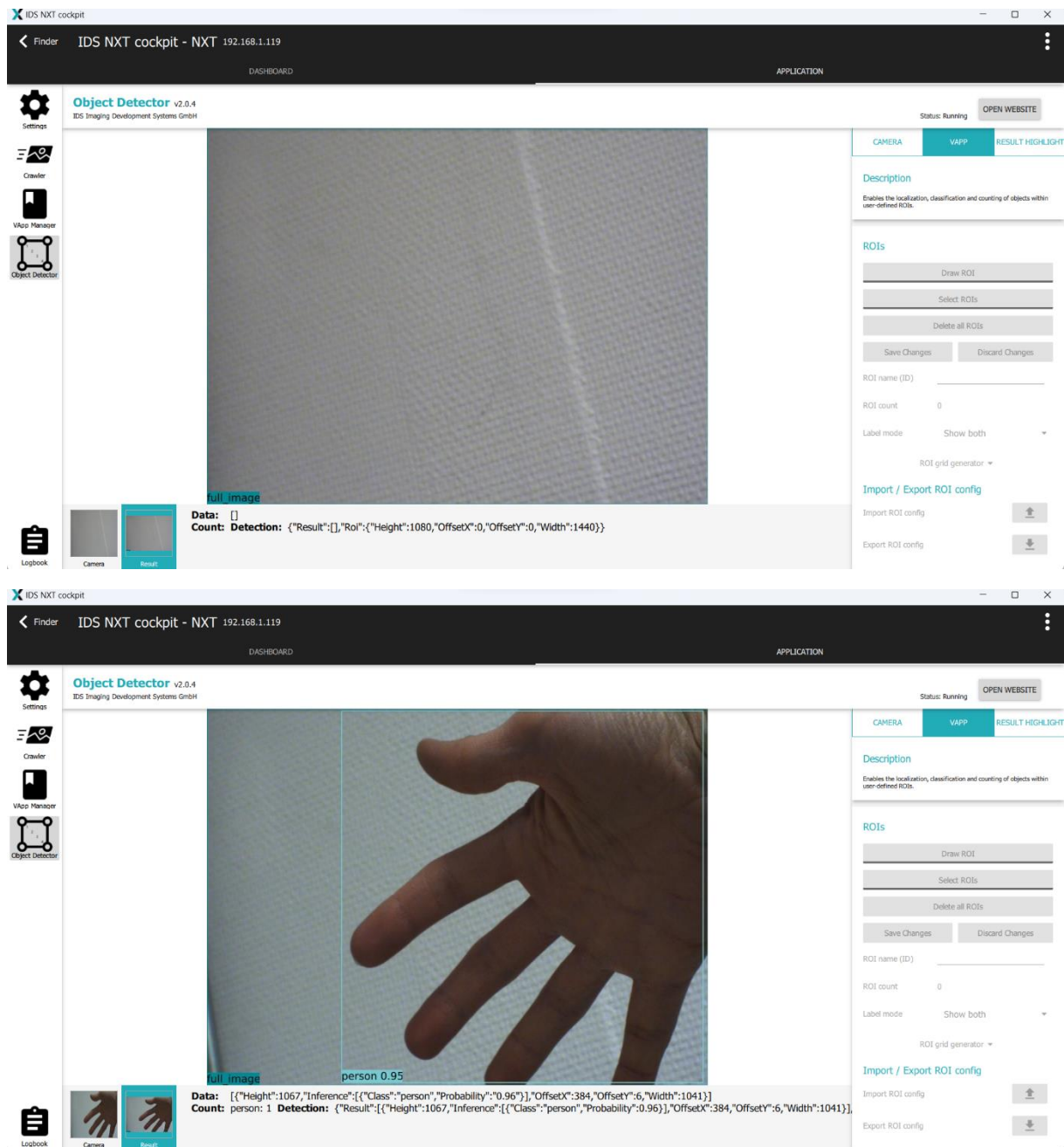


Figure 2: Images from the IDS NXT rio camera with the IDS Convolutional Neural Network Object Detector – PersonDetector. Above: No person in sight. Below: The hand of a person is enough to get a 95 % detection probability of a person.

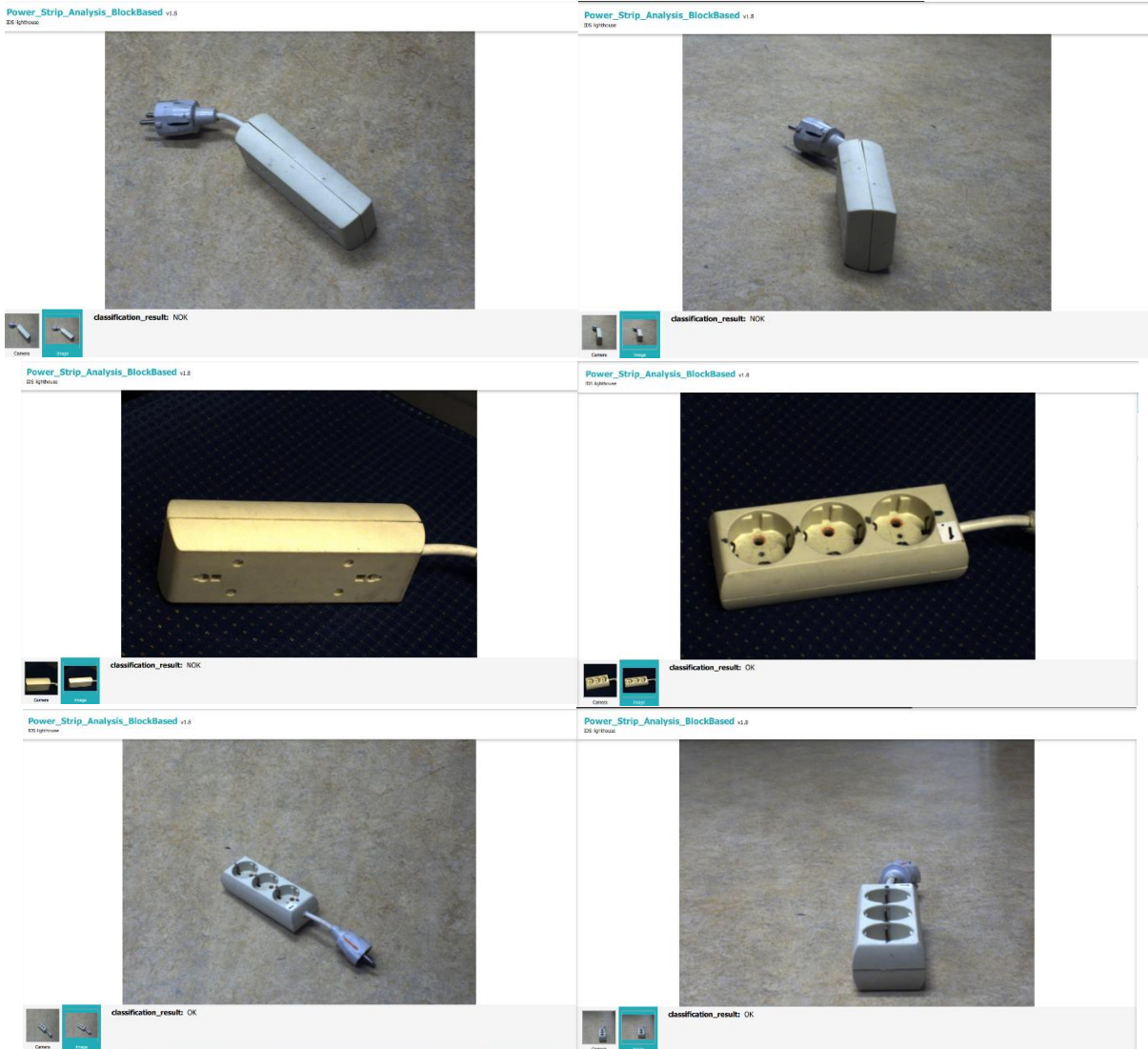


Figure 3: Images from the IDS NXT rio camera with a user created IDS lighthouse Convolutional Neural Network *Power_Strip_Analysis_BlockBased*. This CNN is a very simple OK or not OK model trained to identify power strips with the “top” side seen from the camera. The model does handle this well here, as seen below the camera frames where the result is written in the screen shots, with a CNN created using an input of 141 images. Top: A power strip with the top side turned from the camera. Middle left: A power strip with the top side turned from the camera on a dark background. Middle right: A power strip with the top side visible for the camera on a dark background. Bottom: A power strip with the top side visible for the camera.