WHITE PAPER

WAHTARI

Wahtari nLine Quality Control On-The-Edge

Inline quality control solution using AI for 360 degree surface inspection of tube-shaped products on the edge with just gold samples.

Powered by Intel® Movidius™ Myriad™ X (Myriad X), Intel® Xeon® Scalable Processors (Xeon) and Intel® Distribution of OpenVINO™ Toolkit (OpenVINO).

INTEL® AI BUILDERS MEMBER

Table of Contents:

Introduction

With the rise of Industry 4.0 many companies strive to automate their business and manufacturing processes and make their factories smart. Quality control (QC) plays a major part in every production and is key to how your customers perceive your products and the likeliness they will return to buy more. Unfortunately, it is also one of the most difficult steps to automate, as it still heavily relies on humans to closely examine the products and check for an almost endless list of potential defects and errors, often relying on the experience of the skilled labor to decide, whether the product is still acceptable or must be sorted out. In combination with the fact that it is often not possible to strictly specify every single defect that could occur, automating this step in the production line has proven to be difficult, at least.

Today, greater intelligence is brought to factories in the form of a variety of IoT devices like cameras, sensors or actuators. The constant improvements in the hardware sector makes today's embedded systems smaller and smaller, whilst still increasing their performance.
Naturally, such devices are equipped with networking capabilities and can be connected to the local computer network. This allows to segregate sensitive data from the public Internet and fosters low latency applications. This processing and handling of data at the locations it is generated is called "edge computing", and is an opposing trend to the general cloud approach from before. In addition to the aforementioned advantages, edge devices decouple businesses from the clouds of big tech giants, their subscription fees and the need for a permanent stable Internet connection.

Brief introduction of Wahtari

Wahtari is a German company for computer vision solutions founded in July 2019. Their main focus lies in the area of Artificial Intelligence and end-to-end solutions, meaning that customers receive a complete package to solve their problem, including both soft- and hardware components. Paired with standard API interfaces, the solutions by Wahtari are as simple as it gets. In the relatively short time period since their founding, Wahtari has already successfully implemented a wide variety of different projects, ranging from collecting the bark to wood ratio of logs, detecting cracks, discolorations and deformations on roof tiles to the high-speed recognition of license plates.

Problems with current inline solutions

When an optical analysis of a product is needed to ensure its quality, most of today's visual inspection systems analyze the individual pixels of the captured images by certain rules and clever algorithms to detect specific errors. Certain tool kits are available for this, like the famous OpenCV project, to filter noise, detect edges, calculate rotations, etc. While this approach works very well for setups with perfect, consistent lighting and gray images, a lot of issues occur once the environment starts to change and we have to deal with a lot of noise or different scenarios (like different colors). Naturally, manufacturing facilities grow over time and still contain a lot of older machines and equipment, therefore, the ideal environment is usually either simply non existent, or too costly to realize.

Another issue arises with high speed production lines that may well surpass the 500 m/min mark to up to 1500 m/min. This poses a difficult challenge to the quality control, since faster speeds not only mean larger computational resources are required, the application must also respond faster, since higher latency means that more scrap is being produced.

In addition, the broad error spectra that modern quality control must be able to cope with are very hard to come by with traditional programming, where every possible situation must be taken care of specifically. This leads to larger code bases that are difficult to maintain and extend, and come at higher costs.

Imagine a producer that outputs differently colored (or multi color) power cables, ranging from complete black to complete white with every color in between. In order to support all types, a traditional CV approach would have to deal with an enormous number of different scenarios, since certain algorithms (like adaptive thresholds) have different effects for varying colors. And on top of that, the product might rotate around its x-axis, complicating things even further if it is not uniformly colored.

Finally, current inline solutions tend to need negative samples as examples for the errors they should detect. Based on our experience, this usually introduces further problems. First, the ratio with which scrap is produced by the customer, is most certainly already at a minimum, making it hard to collect any at all. Secondly, customers usually want to also check their products for defects that they have not yet encountered or are not sure happen at all ("theoretical errors"). This requires them to produce, on purpose, junk that fits these criteria, lowering their revenue as production resources are bound.

Overview of *nLine*

nLine is Wahtari's solution to all of the aforementioned problems, revolutionizing the field of inline quality control. Combining traditional computer vision techniques with the latest AI technologies, various features can be detected, such as discolorations, deformations, surface damages, text (OCR) or wrong diameters. It features three cameras for a full 360° inspection and can be easily integrated into existing production lines.
Additionally, this allows customers to buy just one solution from Wahtari, where earlier one for each feature was needed, leading to fewer moving parts in the production line and lower acquisition costs. In the course of this paper, we focus on the technical details of surface inspection.

Hardware setup

Its main core consists of a steel cuboid with two opposing circular holes in the middle, where the tube-shaped product is conducted through. Inside the box, three high speed, full HD cameras are positioned around the channel, with 120 degrees gap between each (see figure 1a-c), to achieve the full 360° coverage.

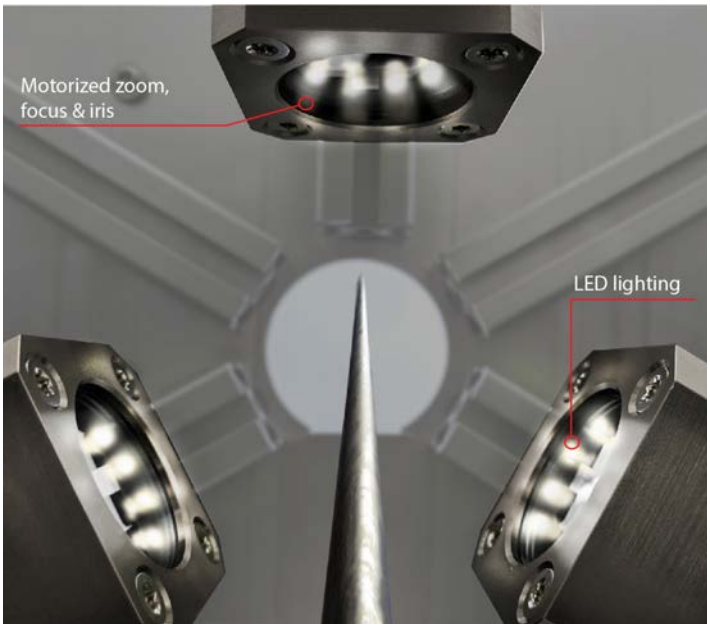

Figure 1a: *nLine* camera setup (1a-c)

Figure 1b

The default image sensors already output up to 160 frames per second (fps) and are connected over USB directly to the local compute unit (called *nBox*), which is positioned underneath the cuboid at the bottom. USB was chosen due to its higher data rate, since most available GigE image sensors only support 1 Gbit/s, whereas USB 3.0 already offers 5 Gbit/s, with its successors in version 3.1 or even 3.2 each doubling this rate. Every camera sits in its own housing and is equipped with a custom controller PCB designed by Wahtari, that features powerful LEDs at the front, making sure the inside of *nLine* is sufficiently illuminated, and several small motors and a low power microcontroller on the back to control the motorized lens (zoom + focus) and adjust the intensity of the LEDs. This enables the workers to easily adjust *nLine* to any kind of setup during production (see figure 3). Due to this modular design and custom housing, it is possible to easily switch out any components, like image sensors or lenses, if the customer's situation should require it.

Figure 1c

Software setup

The core software component is the nLine application running on nBox, programmed in Go and C++/C. It connects with both the cameras and their associated controllers and offers an intuitive UI interface called *vision*, making it possible to view the streams of each camera, adjust their driver parameters and control the application, in general. Users can create products (e.g. a certain red/yellow power cable, or a specific copper tube), which are saved to a small database inside *nLine* and can be activated later, when the product gets produced. Several API interfaces are available, including industry standards like ProfiBus/ProfiNet or TCP/IP based protocols like HTTP with JSON. Customers can then integrate the current status of the ongoing detection, live streams of the cameras and other data into their own applications and processes.

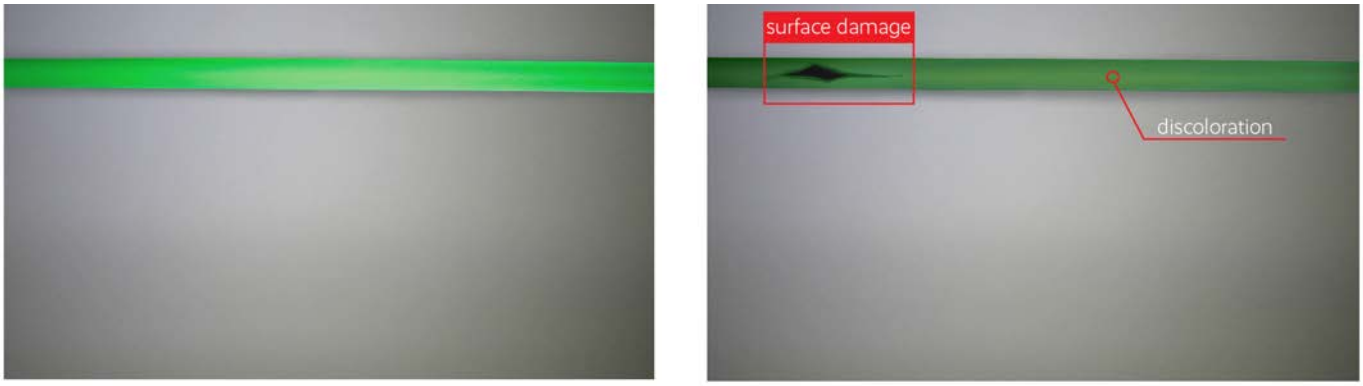

Figure 2: Easy to detect negative production sample

How it works

In order to overcome the issues presented previously, we took a different path than using just traditional CV techniques, but instead combined it with convolutional deep neural networks (CDNN), which are part of the broader deep neural network spectrum.
Artificial Intelligence is a technology that is perfectly suited for these kind of problems, where a definite rule based approach is not possible, or feasible, and a more general, human-like behavior is desired. Instead of programming the exact logic, we want to teach a system to detect and find specific classes of errors or deviations from previously shown reference material.
This dramatically reduces the complexity of the issue, as we no longer need to code out every specific rule, but instead just show the system what it should detect, drastically lowering development time and increasing flexibility of the application. If customers change existing products or introduce new variants in their line, a simple retraining is needed, instead of any code adjustments.

When creating a new product, *nLine* will first collect the so-called "gold samples", by capturing images of the product while it is passing through. Then, a CDNN is trained with these samples, and finally, the model is saved along the product for later usage. It is possible to share the product then with other *nLine* units, since different production lines might produce the same product. Once a product is activated, its model is loaded and fed with the images from the cameras. The model outputs a value that indicates how much the currently captured images deviate from the previouslycollected gold samples, allowing to detect any kind of surface damages, color errors or deformations. Of course, a threshold can be set per product, so that very small deviations (for example due to different surface reflections) do not trigger an error report. Figure 2 shows an example of both a gold and production sample.

Now, this approach might look like it is not much different from simply comparing a reference image pixel by pixel with a captured image. However, this naive approach could not cope with rotations around the x-axis (e.g. multicolor cables) or differently illuminated environments, where the AI model learns to consider and compensate such cases, making it much more resilient and more closely modeling human behavior. Looking for example at figure 3, which shows a positive production sample next to the gold sample, it becomes obvious that the differently illuminated production sample is hard to detect with a simple pixel-comparing or color-clustering algorithm.

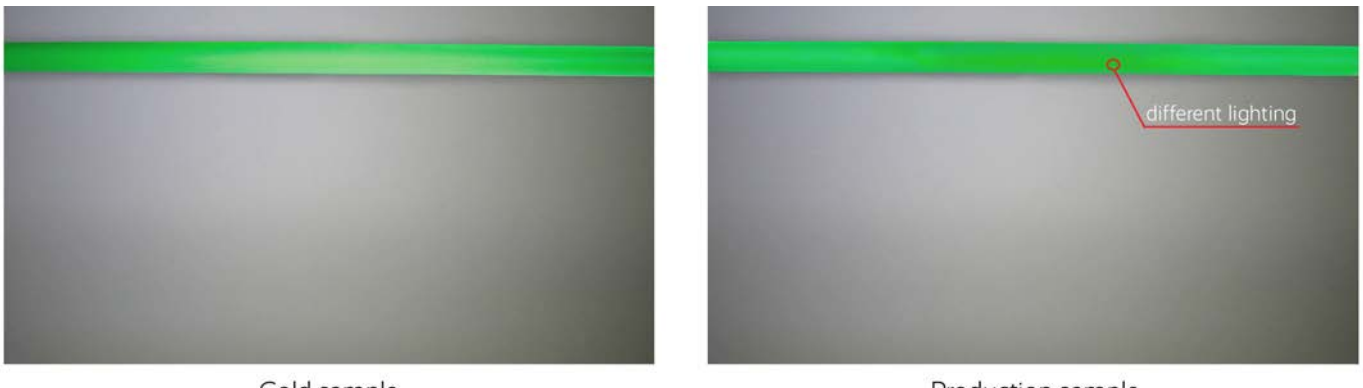

Figure 3: Hard to detect positive production sample

CDNN training with Xeon

Several years prior, deep neural networks (DNN) were mainly being used in the context of cloud computing, since the enormous computation power required to train them could only be provided by powerful GPU server clusters. With today's advance in hard- and software, however, DNN training on the CPU has become feasible.

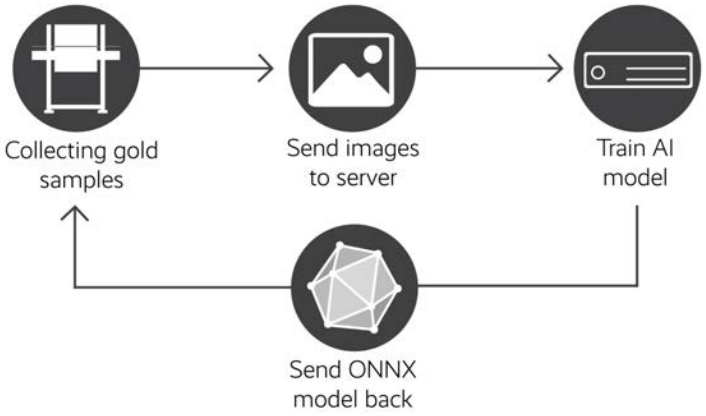

Figure 4: *nLine* training procedure schematic with external Xeon server

Usually, manufacturing facilities or production plants already operate some servers equipped with Intel Xeon processors for their internal IT infrastructure. *nLine* reuses it for training its AI models, which is great to get the maximum value out of the hardware and reduce its idle time.
This works by sending the collected gold sample images to a very small training application running on one of the Xeon servers, where the training loop starts and trains the AI model for several epochs. After it has finished, a simple ONNX model is exported and sent back to nLine, ready to be used. Check out figure 4 to see a graphical representation of this process. Wahtari uses Tensorflow (TF) as AI training framework, along with special Intel optimizations, such as the famous Intel math Kernel Library (Intel MKL) for faster math operations and the powerful Intel® AVX-512 Instructions. An additional improvement was to disable Hyperthreading on the Xeons. Combined, this results in a ~60% faster training process (see figure 5).

CDNN inference with Myriad X

While the training itself is the vastly more computational intensive task, the execution or inference of CDNNs requires quite a few resources as well, especially with *nLine's* combined 480 frames per second output. Here, the Intel Myriad X, a tiny AI accelerator chip that just requires around 2.5W, comes into play, which is able to run most DNNs with an unmatched power to performance ratio. The computation unit *nBox* is equipped with eight of these chips with a combined power usage of roughly 20W.
The images captured by the cameras are preprocessed by *nBox's* Intel Xeon E1275 v6 and then fed into the array of Myriad accelerators. The chips themselves are controlled through the OpenVINO framework. First, the trained ONNX model must be converted with the Model Optimizer tool, which translates each layer to a representation that the Myriad can work with. At the same time, several optimizations are performed to further speed up the inference.
The images can then be run through the loaded network. Our benchmarks show an ~341% / ~194 % inference speedup of the Myriad cluster over the onboard Xeon without / with OpenVINO, when just comparing their raw performance regarding inference-only (see figure 6). In the final application, though, the difference becomes even larger, since the Intel Xeon would have to handle the camera streams and do the post-processing, while occasionally serving requests from users, all on top of the inference.

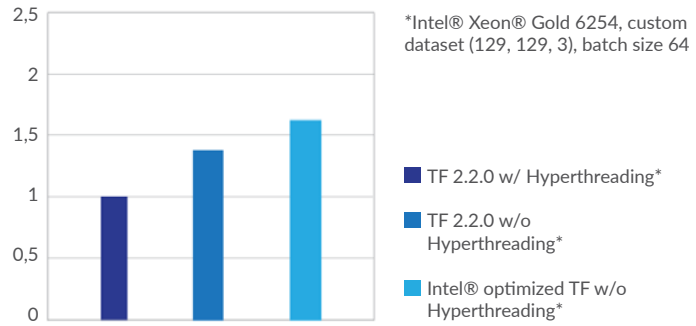

Figure 5: Normalized training performance for avg. epoch duration @ 4000 samples

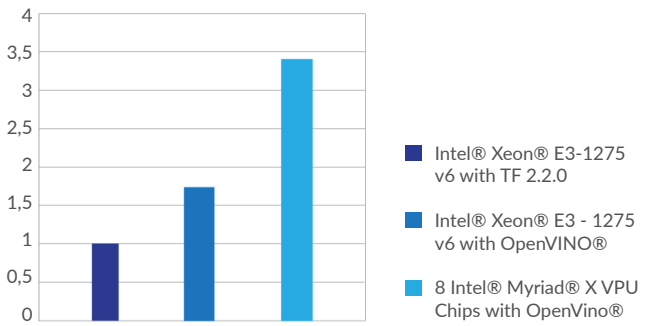

Figure 6: Normalized infence performance for latency and FPS

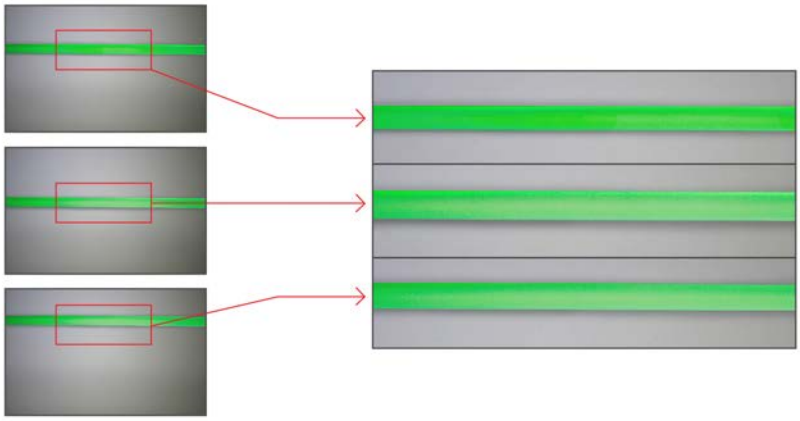

Figure 7: Merge algorithm to create one picture out of three

Optimizations

While this speedup is remarkable, there is still room for optimization. Three high-speed cameras produce a lot of image data per second that need to be analyzed. In order to reduce this number, we are slicing together the three individual camera pictures to one image and feed that into the network, reducing the workload by 2/3. See figure 7 for an example of how this might look. For the AI model itself, it does not matter whether it processes an image with one, two, three or any number of products, as long as the resolution of the individual things it should recognize do not noticeably degrade. It all boils down to the training that uses the same technique and shows the model the merged image of the 3 cameras.

Another optimization, powerful in its effects but simple to achieve, comes in the form of the Async API of the OpenVINO framework. It asynchronously returns the result of the forward pass through the network, allowing the caller to do other tasks in the mean time. This can boost the performance quiet significantly, where our tests have proven this to be especially true for the Myriad devices, resulting in up to double the performance.

Conclusion

Modern hardware and software components paired with special accelerators bring deep neural networks finally to the edge and allow to automate processes in the quality control sector that before relied heavily on human interaction and observation. Wahtari's *nLine* is a powerful example for these kind of solutions and reliably automates inline inspection of tube-shaped products, leveraging the benefits of traditional computer vision and combining them with state of the art convolutional artificial intelligence, while requiring little or no effort from the operator.
Edge solutions have shown great advantages for consumers with privacy, security and reliability increasing, while latency and long-running costs can be severely lowered.

More information:

Wahtari nLine solution:
https://wahtari.io/nline

Wahtari is a member of the Intel® AI Builders ecosystem:
https://builders.intel.com/ai

Intel® technology for Artificial Intelligence:
https://www.intel.com/content/www/us/en/artificial-intelligence/overview.html

INTEL® AI BUILDERS MEMBER

WAHTARI