AI-based Product Quality Controlling in an Anodizing Process

Joshua Prim
Cognitive Information Systems, KITE - Kompetenzzentrum für Informationstechnologie
Technische Hochschule Mittelhessen
Wiesenstr. 14
35390 Gießen
joshua.prim@mnd.thm.de

Henrik Pöschl
Seidel GmbH & Co. KG
Rosenstraße 8
35037 Marburg
henrik.poeschl@seidel.de

Michael Guckert
Cognitive Information Systems, KITE - Kompetenzzentrum für Informationstechnologie
Technische Hochschule Mittelhessen
Wiesenstr. 14
35390 Gießen
michael.guckert@mnd.thm.de

Keywords
Artificial Intelligence, Machine Learning, Quality Assurance, Automation Process Monitoring

ABSTRACT
Product quality is a crucial factor of customer satisfaction and thus directly influences the competitiveness of a company. In manufacturing companies the quality of production processes obviously has significant impact on product quality. Therefore, establishing automated quality control offers considerable leverage for improving processes without necessarily increasing work efforts and costs. In this paper an artificial intelligence based pattern recognition method for increasing the output of an anodising process for aluminium parts is discussed. In the use case presented here, customers have high aesthetic requirements regarding the products which are used in an expensive market segment with only limited fault tolerance. Preparation of the product parts before going through the anodising process is a manual, tedious, and error-prone task that nevertheless requires highest precision. Small deviations can lead to quality problems causing rejections and enforcing repetitions in production. We discuss the application of visual image processing with an artificial intelligence algorithm integrated into the information system of the company to monitor the process and prevent human errors. Results show that our approach reaches high accuracy and can potentially improve delivery reliability with respect to time and quantity by reducing cost-intensive manufacturing errors.

1. PROBLEM DESCRIPTION
Efficiency of processes and quality of products play a decisive role in running economically successful industrial production. Therefore, product quality assurance (QA) is becoming an increasingly important element of the value chain. Seidel GmbH & Co. KG in Marburg is a leading producer of packaging material for cosmetic brands and components for aluminium products. Aluminium foliage is formed coldly in progressive a press and then receives surface treatment in an anodising process.

Before entering the anodising process aluminium parts are placed on racks. Each rack has a capacity of up to 300 parts, depending on the product and the process configuration. Racks are put on carriers which are then moved automatically into the anodising plant. The electrochemical anodising process covers the raw aluminium parts with a layer of aluminium oxide. Colours and varying product appearances can be created.

Furthermore, the aluminium part receives a calloused, scratch-resistant high-quality surface. This complicated process requires product carriers to be loaded strictly according to defined patterns. Due to their size and shape, different aluminium parts require a careful setup of this process.

Depending on the item in production in a given batch, different plug patterns are used to load the product carriers. A plug pattern defines how racks loaded with products have to be placed on the carrier. A product carrier can hold 28 racks (14 on each side of the product carrier), resulting in several million possible combinations for the arrangement of racks on the product carrier. However, only a few plug patterns are feasible in the anodising process and have proven to lead to good results. Fig. 1 shows an example of such a product carrier.

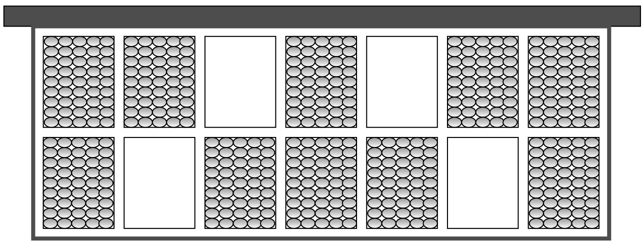

Figure 1: Schematic representation of a loaded product carrier. The round balls symbolise products on a rack which, due to their arrangement in the slots represented by the rectangles, result in the plug pattern on the rack.

Loading product carriers is a manual task in which time pressure and the complexity of the task lead to errors. A typical error is misplacing racks so that the loaded product carrier does not match the pattern required for that particular product. Such a pattern mismatch in the feed of the anodising plant leads to low output quality, so the complete batch has to be rejected. In this case, this batch has to be reproduced incurring additional costs and finally shipping delays.

Therefore, the use of artificial intelligence methods to monitor the loading process of the anodising plant was analysed. Image recognition with a Deep Learning algorithm was implemented to interfere and stop the process in case of loading errors. An independent functional unit (in the following called QA-Engine) was developed, which compares patterns detected in images of the product carrier to patterns that are defined in the master data of the enterprise resource planning system (ERP system). In this paper, we describe the proof of concept for the use of the QA-Engine. We first discuss related work and then present our methodology. The results achieved are then presented. The paper concludes with a look at future work in general and the specific use case presented here.

2. RELATED WORK

In recent years, machine learning methods have been used in many different environments and their fields of application have already been extensively illuminated. Quality control, especially process control, has also been investigated to a certain extent.

One area that has been considered is recognition of patterns such as shifts, trends and cycles in quality control charts with the goal to minimise deviations by identifying the cause and adjusting the process accordingly. Guh and Shiue [Guh and Shiue, 2005] showed that decision trees can be used to identify patterns in control charts. Wang et al. [Wang et al., 2018] also managed to use decision trees to identify anomalies. Gauri and Chakraborty [Gauri and Chakraborty, 2007] successfully trained a neural network to identify features in control charts.

Other studies aimed at determining whether machine learning can be used to classify processes with or without control. Smith [Smith, 1994] used a neural network for X bar R charts to determine whether a process is out of control. For detecting significant shifts in mean values, the researchers were able to achieve equally good results with neural networks as with regular control limits. For small shifts, however, neural networks exceeded conventional control limits. Shao and Chiu [Shao and Chiu, 1999] trained a neural network to identify different assignable causes in an attempt to integrate statistical process control with feedback control for a set of parameters.

Pacella and Semeraro [Pacella and Semeraro, 2007] point out the problem that many quality characteristics correlate and use a neural network to monitor the quality of autocorrelated process data. Low et al. [Low et al., 2003] also consider autocorrelated data, but focus specifically on detecting variations of variance.

Machine learning has also been used to identify anomalies in surface textures ([Weimer et al., 2016]; [Wang et al., 2018]). Methods of image recognition to identify anomalies on textile textures were also investigated ([Ngan et al., 2011]; [Sajid, 2012]).

Plastic injection moulding was also investigated, using various parameters from production as input ([Ribeiro, 2005]; [Tellaeche and Arana, 2013]). Zhao et al. [Zhao et al., 2017] describe an automatic image recognition approach for quality assurance for cold rolling processes. Villalba-Diez et al. [Villalba-Diez et al., 2019] have applied machine learning in the printing industry. The researchers show how a neural network can be combined with a high-resolution optical quality control camera to increase product quality and reduce costs in the printing industry. Ferguson et al. [Ferguson et al., 2018] on the other hand, demonstrate the use of neural networks to identify casting defects in X-ray images.

The fast-food chain Domino's Pizza is also using machine learning to monitor the quality of its pizzas. In 2019, the company introduced a scanning technology in its kitchens in Australia that uses machine learning to analyse images of pizzas. According to Domino's, it has succeeded in increasing the quality of the pizzas monitored in this way by fifteen per cent ([Dominos, 2019]). The US company eBay Inc. also relies on machine learning for quality assurance. eBay uses a neural network to classify whether a UX component meets the desired quality criteria or not (see [Sharan et al., 2018]).

3. METHODOLOGY

For error detection, we use a convolutional neural network (CNN) architecture. In this section we will at first provide some theoretical background about CNNs in 3.1 and then briefly describe our our database structure in 3.2.

3.1 Convolutional neural networks

CNNs ([LeCun et al., 1989]) represent a special variant of artificial neural networks (ANN). Due to their structure, they are particularly well suited for image recognition and are preferably used for the classification of images and videos ([Schwaiger and Steinwendner, 2019]). The CNN model was inspired by the mechanisms of the visual cortex of the brain. A significant difference to conventional ANNs is that CNNs apply filters and create feature maps to detect patterns and structures in images. These can be contours, colours or textures, which are then combined into more complex structures.

Architecturally, the structure of CNNs shows specific differences compared to conventional ANNs. The input layer usually takes three-dimensional input in the form of the spatial extension of the image (width * height) and has a depth representing the colour channels (usually three for the RGB colour channels). This is followed by convolution layers and pooling layers, explained below. These two types of layers can be repeated with tailored parameterisation. Typically, a classification layer is used as output layer, represented by a fully linked layer for generating the scores. A high-level general CNN architecture is shown in Fig. 2.

- Input: The CNN input is usually the 3-channel colour image or 1-channel grey image matrices, containing the intensity values at each position.
- Conv: The core of a CNN is the convolutional layer. This layer performs the mathematical operation called convolution. Convolution is a special kind of linear operation. CNNs are thus ordinary neural networks that use convolution instead of general matrix multiplica-

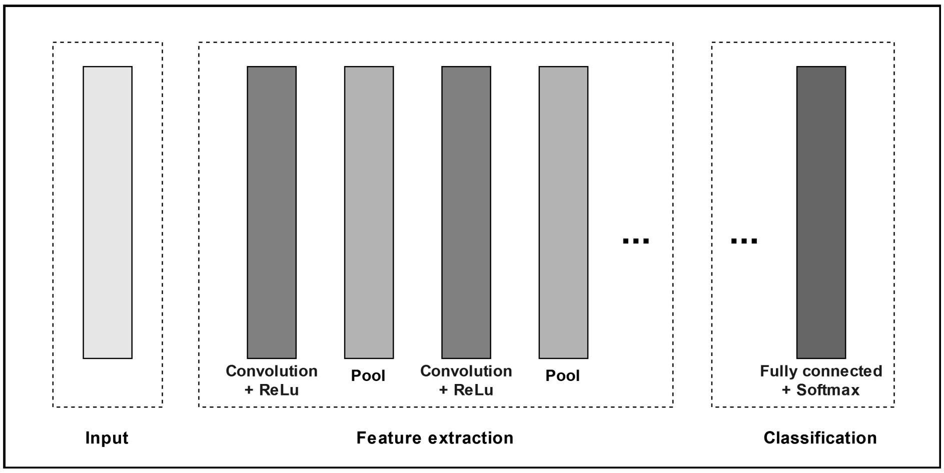

Figure 2: General CNN architecture

tion in at least one of their layers ([Goodfellow et al., 2016]).
During convolution, the size of the filter (kernel size) (e.g. 3 x 3) is defined first. Then the filter scans the pixel matrix of the input like a window with a constant step size. The filters move from left to right across the input matrix and jump to the next lower row after each pass. The so-called padding determines how the filter should behave when it hits the edge of the matrix. The filter has a fixed weight for each point in its viewport, and it calculates a result matrix from the pixel values in the current viewport and these weights. The result of the convolution is called a feature map in the terminology of CNNs. The size of this resulting matrix depends on the kernel size of the filter, the padding and especially on the step size. A non-linear activation function is used for the feature map. For modern CNNs, the default activation function is the rectified linear activation function (ReLu).

- Pooling: The pooling layer leads to a reduced spatial dimension by using the pooling function according to Zhou and Chellappa [Zhou and Chellappa, 1988]. It aims to reduce the amount of network parameters and the calculation costs. The pooling layer is often placed between two successive convolution layers. Different aggregation functions can be used for pooling. The most common aggregation functions are max pooling and average pooling ([Lin et al., 2013]).
- Dense: The Fully Connected Layer or Dense Layer is a standard neural network structure in which all neurons are connected to all inputs and all outputs. Moreover, the last fully connected layer produces the output of the entire net. Each value of the k-dimensional output represents the probability of the corresponding label using the softmax function.

Combined, these layers provide a complete CNN into which input can now be fed for network decision-making.

3.2 Dataset

Images taken in the anodising plant serve as the data basis for the QA-Engine. For this reason, a system for image acquisition was integrated into the anodising process, which records the pattern set by the employees. The system uses an RPI3-CM01 camera to take images of the product carrier with the racks at the start of the anodising process. Such an image is shown in Fig. 3.

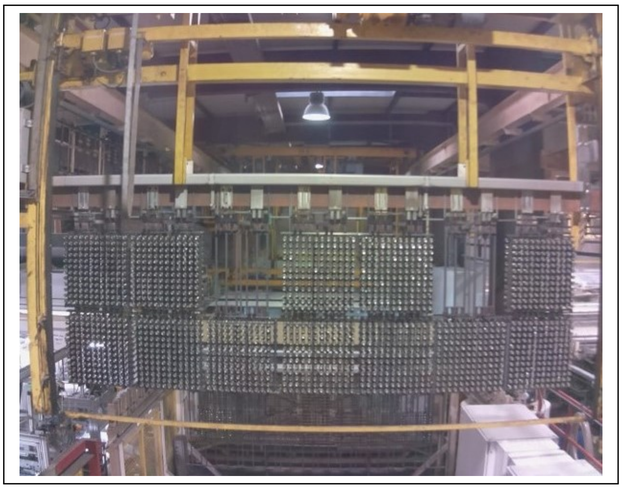

Figure 3: Example of an image taken in the anodising plant

After the product carrier has been completely loaded with the racks by the machine operators, it is automatically moved through the anodising system. While the product carrier is moving through the anodising plant to the preprocessing, an image is taken automatically triggered by an ultrasonic sensor of the type HC-SR04. Thus, images are always taken at the same point and all have identical distance from the camera. Additional information of the current order is required to evaluate the pattern. For each new order, information is captured by the ERP system and stored together with the information which plug pattern is used for the order. The information describing the orders is delivered via an API from the ERP system. In preparation for our experiments more than 20000 images were reviewed with low-quality images eliminated. The remaining rest was labelled with a plug pattern resulting in a data set of 1000 labelled images (elox_seidel_1k). When creating the data set, care was taken to ensure that the distribution of images per plug pattern was balanced. This data set is used for the training of the CNN and for all experiments carried out. The elox_seidel_1k dataset was then further processed to be suitable for the training of the CNN. The QA-Engine responsible for data preparation reads the data set and processes the data.

In these and further steps, an X-data set and a Y-data set are created, representing input and target values. The X-data set consists of an array of the form (1000, 256, 320, 3). The first value represents the number of image instances. The second and third values are the dimensions of the image. The fourth value represents the three colour channels of the image (RGB). The Y-data set corresponds to the form (1000, 10), whereby the first value also represents the number of image instances. The second value specifies the plug pattern class in an one-hot-encoding. Finally, the image instances are divided into training, validation and test data set. The data records can now be used to train and validate the CNN.

4. QA-ENGINE

In this section, the architecture of the QA-Engine is described, and the CNN implementation is discussed.

4.1 The QA-Engine in the anodising plant

The architecture of the QA-Engine was developed so that it can be combined with the existing components of Seidel GmbH & Co. KG and can easily be integrated into the system landscape of Seidel. See Fig. 4 for an overview of that landscape.

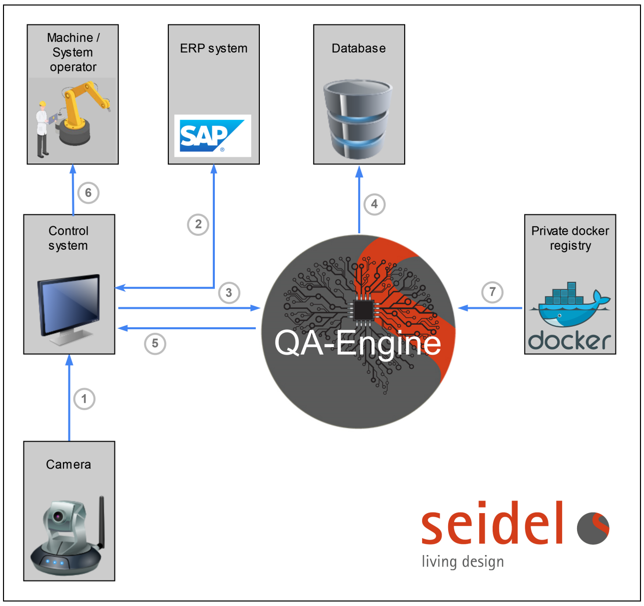

Figure 4: Integration of the QA-Engine into the Seidel environment

The architecture is designed to support the quality assurance process in the anodising plant in the best possible way. Initially an image of the product carrier with the racks is taken in the anodising plant. This image is then forwarded to a control system in the anodising process [1]. The plant control system asks the ERP system for the order currently in the anodising plant [2]. The recorded image, together with the current order from the ERP system, is sent by the plant control system to the QA-Engine [3]. The QA-Engine analyses the data and returns an assessment of the plug pattern to the plant control system [5]. It is also planned that the QA-Engine will store generated assessments in a database for potential further evaluation in the future. [4]. The results database created by the QA-Engine is forwarded by the system control if required. If the QA-Engine detects an error in the plug pattern, the result is forwarded to the system operator [6]. It is subjected to a human who can, if necessary, intervene in the anodising process. In terms of a fully automated production, the result of the QA-Engine can also be forwarded directly from the system control to the machine in the anodising plant. As a result, the machine interrupts the process and the plug pattern can be corrected. The fully implemented QA-Engine was provided as a docker image in a private Seidel registry [7].

Communication with the QA-Engine is done via a representational state transfer application programming interface. This enables interaction with the system using hypertext transfer protocol requests. The system control of the anodising plant sends a request to the QA-Engine and receives a response from the QA-Engine. Various endpoints have been implemented for communication with the QA-Engine API. Among other things, the images can be sent to the engine and hyperparameters can be requested and adjusted.

4.2 Model Implementation

The main component of the QA-Engine for quality assurance is a CNN. Based on the pictures taken, the CNN recognises the plug pattern depicted in the anodising plant. This way classification of the images and thus of the plug patterns is to be carried out. One class represents one plug pattern in the anodising process. The CNN was implemented in the programming language Python (version 3.6) (cf. [van Rossum and Drake Jr, 1995]). TensorFlow (Version 1.9) ([Abadi et al., 2016]) was used for this purpose. TensorFlow is an open-source program library for machine learning. TensorFlow combines the computational algebra of compilation optimisation techniques and thus facilitates the calculation of many mathematical expressions. Time-consuming calculations can thus be processed much faster ([Zaccone, 2016]). Keras ([Chollet, François, 2015]), an interface for TensorFlow, was also used for training and validation of the model.

Different architectures were implemented and assessed to determine the network architecture most suitable for the problem. Based on the results of the comparison, the architecture shown in Fig. 5 was selected. As shown in Fig. 5, the network consists of nine layers with weights; the first three are folded and the remaining six are for regulation, as well as fully connected. The output of the final fully connected layer is fed into a 10-way softmax, which creates a distribution over the ten class labels.

The first convolution layer filters the $256x320x3$ input image with 32 filters with a $5x5$ window and a stride of one pixel.

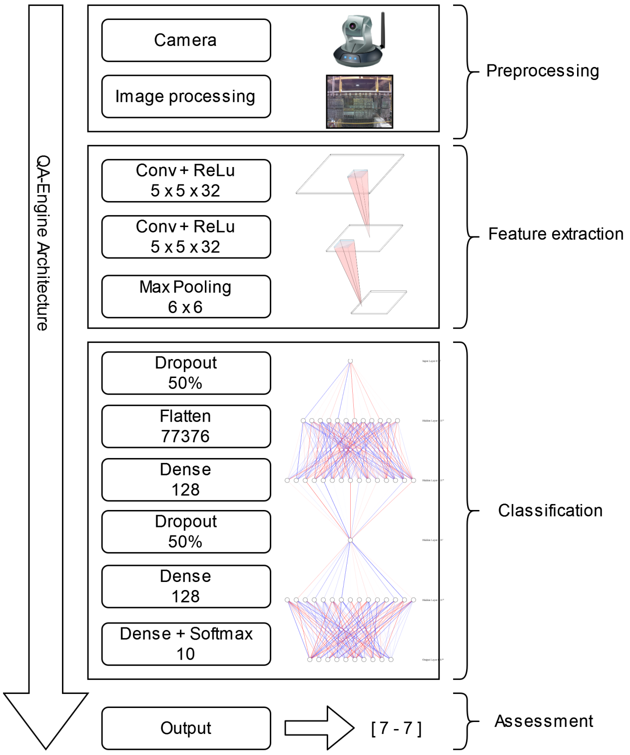

Figure 5: Visualisation of a schematic representation of the architecture of the QA-Engine.

The second convolutional layer takes the output of the first convolutional layer as input and also applies 32 filters with a $5x5$ window. The two layers are followed by a pooling layer with a window size of $6x6$. Padding is not used in either layer.

There is a flatten layer between the convolutional and the fully connected layer. Flattening converts the two-dimensional matrix of features into a vector for processing by the subsequent fully connected layers. The fully connected layers consist of 128 neurons. To prevent the CNN from overfitting, dropout was used. The softmax function underlying the final dense layer transforms the transferred values. The output of the softmax function corresponds to a categorical probability distribution and thus indicates a probability of belonging to a plug pattern.

4.3 Further experiments

Furthermore, the influence of contrast, brightness and sharpness on the CNN used was investigated. On the one hand, this makes it possible to determine whether an adjustment of contrast, brightness and sharpness during the preprocessing of the images possibly leads to better results. On the other hand, the robustness of the network for the factors will be investigated.

For this experiment, the preprocessing was adapted and the network was then trained and tested, using the previously described dataset. Contrast, brightness, and sharpness was manipulated for each of the 1000 images (800 training, 100 and 100 test). Subsequently, the CNN was trained. Nine training and test runs were performed for each factor, each with different value assignments. In addition, the vulnerability of the CNN of the QA-Engine was tested under poor conditions. For example, dust or other impurities may collect on the camera lens when used in the anodising plant. Also, lamps in the production hall may fail temporarily. These and other situations lead to different conditions when the image is captured. Therefore, it was investigated how well the mesh performs with changes in contrast, brightness and sharpness. Therefore, 100 images were randomly taken for this purpose and deliberately manipulated. They were then given to the CNN of the QA-Engine for assessment.

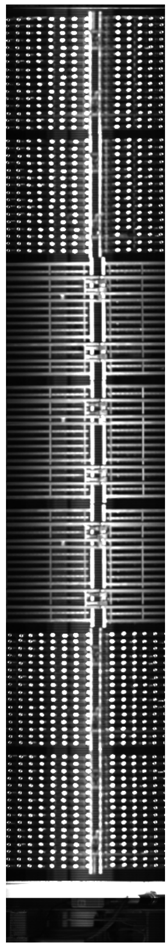

Figure 6: Example image taken with the line scan camera

The possibility of generating patterns dynamically at runtime was also investigated to react more flexibly to the introduction of new patterns. For this purpose, the recorded images were divided into individual quadrants, which were checked separately for a loaded or unloaded slot. A modified test setup using a line scan camera was used for this purpose. The line scan camera is triggered by a incremental encoder when the product carrier is moved into the anodising system. An example of a recorded image is shown in Fig. 6. Since this is an experimental setup, only the central area of the container was recorded. 14 quadrants were defined on the image, which were checked for an existing rack. At runtime, the plug pattern can be dynamically generated depending on whether a rack with aluminium parts needs to be on the quadrant or not. A modified version of the CNN described above was trained to recognise and distinguish between existing and non-existing racks. The network has an accuracy of 98,99%, but the static definition of the quadrants proves to be error-prone in the case of shifted images. Since a fine adjustment of the encoder can solve this problem, this approach will be further investigated in the future.

5. RESULTS

The CNN we created was trained using the previously described data set. Our model was trained on a NVIDIA GeForce GTX 1080Ti 11 GB. 10-fold cross validation was used. The network has an accuracy of 98.09%. The results show that our model can recognise plug patterns and differentiate between plug patterns.

During the experiments, it was found that varying contrast, brightness, and sharpness in preprocessing did not positively affect the predictive ability of the CNN. Tendentially, an extreme adjustment of these variables led to worse results. Further investigations showed that the CNN of the QA-Engine is not very resistant to changes in brightness and contrast. Among other things, the predictive ability deteriorates significantly when brightness is lowered. Changes in image sharpness, on the other hand, are harmless.

6. CONCLUSION AND FUTURE WORK

In this work, we have described the prototypical development of a system for quality assurance. The system was developed for use in the anodising process of the company Seidel GmbH & Co. KG to monitor this process automatically. Thereby, images of product carriers containing racks with production goods arranged in a plug pattern are automatically recorded. These serve as input for the system. The current order from the ERP system is also transferred to the QA-Engine. The image is recognised and classified using a previously trained artificial neural network. For this purpose, the image with the product carrier is assigned to a pattern. The plug pattern classified by CNN is compared with the intended plug pattern from the ERP system. In case of an incorrect plug pattern, the system gives feedback. Errors occurring in the anodising can be detected early thus reducing quality assurance costs.

Currently a fully usable prototype is available which can be integrated into the infrastructure with container virtualisation. Further tests are required and camera technology will be consolidated.

The prototype created in this work has a potential high long term impact to significantly improve the anodising process. Additionally, the prototype is considered to have the potential to be transferred to other use cases, e.g. high precision determination of the loss of products. Individual aluminium parts tend to detach from the racks at odd times and remain in the anodising plant. The QA-Engine can be extended to count the aluminium parts on the individual racks. In this case, images would have to be taken at start and end of the anodising process which could then be used to compare counts giving the exact loss of products. However, this use case could not yet be implemented as camera technology with sufficient precision was not available. Later versions of the QA-Engine can implement a higher degree of automation by independently interrupting processes and initiating reloading of the carriers.

In conclusion, with implementing the QA-Engine for quality assurance Seidel GmbH & Co. KG has made a step towards fully automated and monitored production processes.

REFERENCES

[Abadi et al., 2016] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *https://arxiv.org/pdf/1603.04467.*

[Chollet, François, 2015] Chollet, François (2015). Keras. https://keras.io/, last updated 25.05.2022.

[Dominos, 2019] Dominos (2019). Dom pizza checker. https://dompizzachecker.dominos.com.au/, last updated 21.05.2022.

[Ferguson et al., 2018] Ferguson, M. K., Ronay, A., Lee, Y.-T. T., and Law, K. H. (2018). Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *Smart and sustainable manufacturing systems*, 2.

[Gauri and Chakraborty, 2007] Gauri, S. K. and Chakraborty, S. (2007). A study on the various features for effective control chart pattern recognition. *The International Journal of Advanced Manufacturing Technology*, 34(3):385–398.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Massachusetts and London, England.

[Guh and Shiue, 2005] Guh, R.-S. and Shiue, Y.-R. (2005). On-line identification of control chart patterns using self-organizing approaches. *International Journal of Production Research*, 43(6):1225–1254.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

[Lin et al., 2013] Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400.*

[Low et al., 2003] Low, C., Hsu, C.-M., and Yu, F.-J. (2003). Analysis of variations in a multi-variate process using neural networks. *The International Journal of Advanced Manufacturing Technology*, 22(11):911–921.

[Ngan et al., 2011] Ngan, H. Y., Pang, G. K., and Yung, N. H. (2011). Automated fabric defect detection—a review. *Image and Vision Computing*, 29(7):442–458.

[Pacella and Semeraro, 2007] Pacella, M. and Semeraro, Q. (2007). Using recurrent neural networks to detect changes in autocorrelated processes for quality monitoring. *Computers & Industrial Engineering*, 52(4):502–520.

[Ribeiro, 2005] Ribeiro, B. (2005). Support vector machines for quality monitoring in a plastic injection molding process. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(3):401–410.

[Sajid, 2012] Sajid, T. (2012). Fabric defect detection in textile images using gabor filter. *IOSR Journal of Electrical and Electronics Engineering*, 3(2):33–38.

[Schwaiger and Steinwendner, 2019] Schwaiger, R. and Steinwendner, J. (2019). *Neuronale Netze programmieren mit Python*. Rheinwerk Computing. Rheinwerk Computing, 1. auflage edition.

[Shao and Chiu, 1999] Shao, Y. E. and Chiu, C.-C. (1999). Developing identification techniques with the integrated use of spc/epc and neural networks. *Quality and Reliability Engineering International*, 15(4):287–294.

[Sharan et al., 2018] Sharan, Y., Wang, H., and Rath, S. (2018). Gui testing powered by deep learning.

[Smith, 1994] Smith, A. E. (1994). X-bar and r control chart interpretation using neural computing. *The International Journal of Production Research*, pages 309–320.

[Tellaeche and Arana, 2013] Tellaeche, A. and Arana, R. (2013). Machine learning algorithms for quality control in plastic molding industry. *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4.

[van Rossum and Drake Jr, 1995] van Rossum, G. and Drake Jr, F. L. (1995). *Python tutorial*, volume 620. Centrum voor Wiskunde en Informatica Amsterdam.

[Villalba-Diez et al., 2019] Villalba-Diez, J., Schmidt, D., Gevers, R., Ordieres-Meré, J., Buchwitz, M., and Wellbrock, W. (2019). Deep learning for industrial computer vision quality control in the printing industry 4.0. *Sensors (Basel, Switzerland)*, 19(18).

[Wang et al., 2018] Wang, T., Chen, Y., Qiao, M., and Snoussi, H. (2018). A fast and robust convolutional neural network-based defect detection model in product quality control. *The International Journal of Advanced Manufacturing Technology*, 94(9):3465–3471.

[Weimer et al., 2016] Weimer, D., Scholz-Reiter, B., and Shpitalni, M. (2016). Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals - Manufacturing Technology*, 65(1):417–420.

[Zaccone, 2016] Zaccone, G. (2016). *Getting started with TensorFlow: Get up and running with the latest numerical computing library by Google and dive deeper into your data!* Community experience distilled. Packt Publishing Ltd, Birmingham, UK.

[Zhao et al., 2017] Zhao, Y. J., Yan, Y. H., and Song, K. C. (2017). Vision-based automatic detection of steel surface defects in the cold rolling process: considering the influence of industrial liquids and surface textures. *The International Journal of Advanced Manufacturing Technology*, 90(5):1665–1678.

[Zhou and Chellappa, 1988] Zhou, Y. and Chellappa, R. (1988). Computation of optical flow using a neural network. In *IEEE International Conference on Neural Networks 1993*, pages 71–78 vol.2, Piscataway. IEEE.

Acknowledgments

This research was partially supported by Seidel GmbH & Co. KG and by Hessen Agentur LOEWE 3 (project IPDU - HA project no. 593/18-16)

Contact

For more information contact:

Joshua Prim
Cognitive Information Systems, KITE -
Kompetenzzentrum für Informationstechnologie
Technische Hochschule Mittelhessen
Wiesenstr. 14
35390 Gießen
joshua.prim@mnd.thm.de