Tile Quality Detection Device: Internet of Things (IoT) Demonstration Prototype

Bernhard Heiden[1,3][a], Danijel Pačnik[1], Jessica Pregl[1], Maximilian Müller[1], Simone Leitner[1], Volodymyr Alieksieiev[2][b] and Bianca Tonino-Heiden[3][c]

[1]*Industrial Engineering and Management Studiengang, University of Applied Sciences, Europastrasse 4, 9524 Villach, Austria*

[2]*Institute of Production Management and Technology, Hamburg University of Technology, Denickestrasse 17, 21073 Hamburg, Germany*

[3]*Institute of Philosophy, University of Graz, Heinrichstrasse 26/V, 8010 Graz, Austria*

Keywords: Tile Quality Check, IoT-Prototype, Matlab, Arduino UNO, Microcontroller, Pattern Detection.

Abstract: In this paper we present the development of a quality tile detection device for the Internet of Things (IoT) demonstration in a prototype for educational purposes. For this an Arduino UNO microcontroller is used, and the prototype is chosen after an initial design process, together with suitable tile parts, that imitate possible real tiles in a production line in the downscaled prototype. The Artificial Intelligence (AI) task of pattern detection is performed by a Matlab script. Finally, the Arduino, that controls the conveyor belt and the switch of the tile quality detection device is linked to the Matlab Script, that controls the picture detection and AI-script evaluation on the PC and its backcoupling to the Arduino, yielding a semi-automatic tile quality checking procedure with a human in the loop.

1 INTRODUCTION

A typical application in tile production industry is to test the quality of tiles, as in the production process imperfections can occur (see, e.g. (Frei, 2017)), and have then to be detected in a mass production process. A case, with regard to ceramics industry, which is comparable to our case, is given in (Zhang et al., 2022), showing its relevance and state of the art problem in the industry. This hence can be regarded as an industrial application case where Artificial Intelligence (AI) techniques can be applied, as well as Internet of Things (IoT) applications that implement these tasks in a prototype that merges the cybernetic tasks with the necessary AI for detection.

(Dong et al., 2022) have developed an industrial solution prototype for mosaic tiles as a complete workflow. The system installed on a conveyor belt, has a detection unit, and a sorting unit by means of a spray mechanism.

There exist a lot of AI algorithms that are used for tile quality detection. E.g., that of (Wan et al., 2022) is based on deep learning.

IoT applications range from a narrow field of basic implementations, to a wide field of broad and even global applications, where the focus is then to typical challenges like how to merge these IoT devices to reach one goal (e.g. (Lemoine et al., 2020) and (Choo et al., 2018)), to reach cybersecurity (see, e.g. (Corallo et al., 2022) and (Lindström et al., 2018)), and many others. This means that the tasks, which are performed by IoT-devices, are tremendously increasing with the uprise of more and more miniaturised devices in all industries. The overarching development is triggered by the high amount of microprocessor driven processes world-wide, and the ongoing trend to make those automatisations cheaper and of higher applicability due to the development towards a higher functional performance, e.g. in the mobility and energy savings sector.

For the IoT applications one widely used microcontroller board is the Arduino UNO. It is widely applied for IoT prototyping applications according to its low costs among many other microcontrollers. A good reference for the Arduino UNO is given in (Borchers, 2013).

[a] https://orcid.org/0000-0001-8324-6505
[b] https://orcid.org/0000-0003-0792-3740
[c] https://orcid.org/0000-0001-7648-2833

Heiden, B., Pačnik, D., Pregl, J., Müller, M., Leitner, S., Alieksieiev, V. and Tonino-Heiden, B.
Tile Quality Detection Device: Internet of Things (IoT) Demonstration Prototype.
DOI: 10.5220/0012634000003705

In *Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS 2024)*, pages 209-215
ISBN: 978-989-758-699-6; ISSN: 2184-4976

The computer algebra and programming software Matlab is frequently used in science and industry for solving engineering tasks, and especially it can serve practically to implement AI applications for at least three reasons. Firstly (1) there exists a broad international industrial user community, especially in the area of academia, where a lot of AI-algorithms are developed with this language, (2) secondly there is a good support possible, which is important for industrial applications, and thirdly (3) there exists a wide range of tutorials and ready to use libraries that allows to implement AI-algorithms from the scratch to the complete set-up. E.g., in (Hanzaei et al., 2017) a pattern detection problem is also implemented with Matlab, as in our case for the tile quality detection.

An overview of Computer Numerical Controlled (CNC) machines is given in (Kief, 2015). These are used in mechanical workshops, and are often to be found in prototyping labs or Fablabs (see, e.g. (Gershenfeld, 2006)). To produce a prototype there or in similarly structured workshops, the rapid prototyping approach is state of the art, which means that a series of CNC devices is used to fabricate the prototype, beginning with form construction and ending with machining, e.g. with CNC-milling or 3D-printing (see, e.g. (Gibson et al., 2015)).

One reviewer asked to answer the question <What is new in the paper?>. This paper deals with a new prototype used for education to learn new and actual technologies used in industry and production. (Mesch, 1985) has pointed out that in education new measurement technologies lead also to a new kind of education. (1) New material, (2) new or until now not utilized physical effects, and (3) new applications are, according to Mesch, elements of newness, that induce possible different approaches in teaching as a consequence. Applied to our case, the newness is constituted by the here presented student's project, that has conceptualised the problem of tile quality detection in integrating all the aspects of the state of the art industrial technologies: CNC-production, IoT prototyping and algorithmic or in general AI-open IoT-implementation.

The prototype can now additionally be used for applying industrial concepts in a classroom environment, and it is available as an open source application (Heiden et al., 2024) for the IoT community.

Goal of the Work. The goal of the work is to develop, build and demonstrate an IoT-prototype for an educational classroom setting of a realistic tile producing ceramics industry exercise and as an educational task to combine mechanical engineering tasks with the design of an IoT device, and providing an AI method of pattern detection, by an integrated prototype.

Research Questions. The main research question of the work is: <How can a downscaled prototype of an IoT-device for educational demonstration purposes be built for the quality detection of tiles by applying AI-algorithms in ceramics industry?>.

Content. In this paper we first give in Section 2 an overview over the design of the IoT model tile quality detection device. In Section 3 we discuss the results of this work. In Section 4 we then give the conclusions and outlook of the work.

Methods Used. The method of this work is to design, construct, build and test an IoT model prototype, which incorporates essential elements of an automatised tile quality detection. For the construction a mechanical workshop is used, as well as CNC milling. For the quality detection AI algorithms are implemented in Matlab. The Arduino UNO is used as basis IoT-backbone for the prototype. A half-automatised communication between the Arduino and the Matlab computer application is then used to implement the pattern detection, by means of a human-machine interaction.

2 TILE QUALITY DETECTION DEVICE

In Figure 1 we see the prototype of the tile quality detection device. It consists of downscaled round tile-models, the slide, the conveyor belt the photoelectric and the tile detection sensor and a switch for choosing the right tile, powered by a stepper motor. The flow diagram can be seen in Figure 3.

2.1 Material for Virtual Tile Detection

Since dealing with normal tiles is quite unwieldy, 'sample tiles' are designed and made. Therefore, the patterns are milled from a phenolic resin coated plywood in different variants. The coating is brown, further the coating is glossy and therefore simulates a regular tile surface. The plate thickness is 10 mm. In addition, the tiles have a diameter of 70 mm. The round shape is chosen for the sake of simplicity because we are implementing a model production system. The following samples were produced:

- Model 1: Error-Free

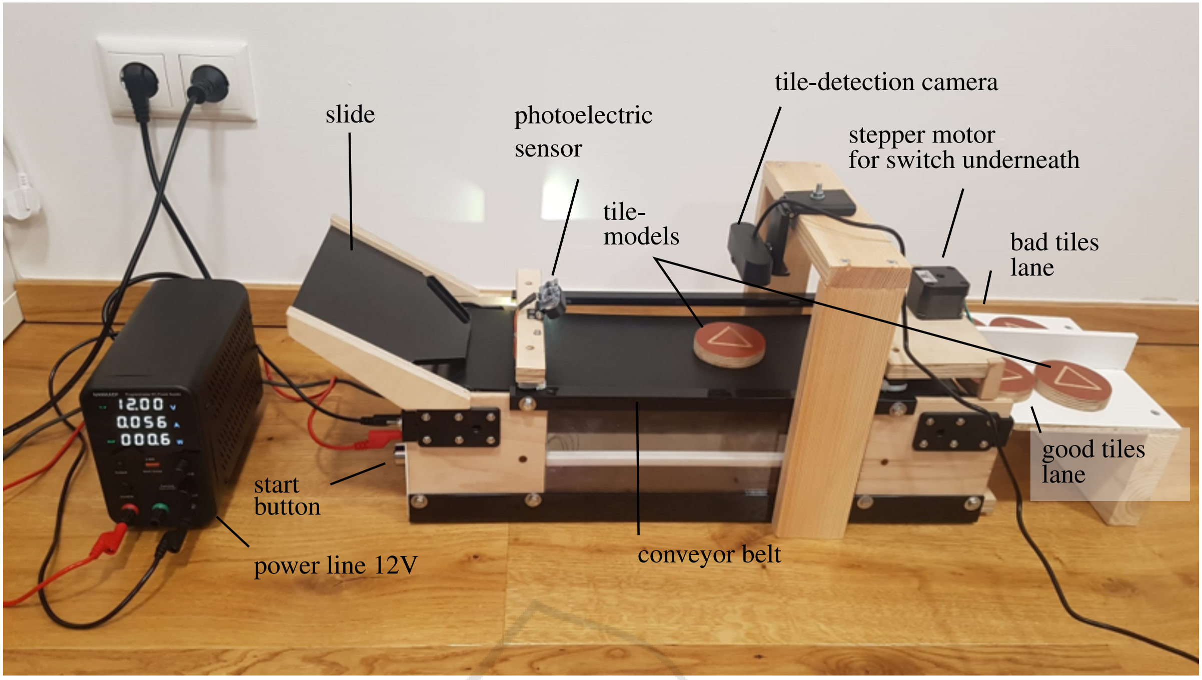

Figure 1: Tile Quality Detection Device IoT-Demonstration Prototype. More details can be found in the complete project documentation (in German) of the prototype in (Heiden et al., 2024).

- Model 2: Error 1
- Model 3: Error 2
- Model 4: Special motive 1 ‘Platypus’
- Model 5: Special motive 2 ‘Koi’

These five Models were chosen to be programmed at the image processing (pattern recognition) in the further course. Model 1 should be recognized without errors, Models 2 and 3 should be recognized with errors and the accuracy of the system should also be tested. The special motives koi and platypus must be recognized as mistakes in Model 4 and 5 (see also Figure 2).

First, the five design Models were each individually converted into 3D drawings/3D models in the program Fusion 360 and the associated G code (see, e.g. (Kief, 2015)) was created. The G code, also called DIN code, is the machine language with which the user controls the CNC machine and with which the CNC machine moves to the desired position or depth and mills or engraves. Secondly, the program Estlcam V11 processes the G code and controls the CNC machine used. Thirdly, the coordinate system of the CNC machine must be determined, and the workpiece must be clamped. Finally, the desired motive is milled.

The used tools were (1) the engraving graver with a 30° tip and an end mill. For safety reasons, the samples were milled beginning with a bar, so that they are not loose in the machine. For this reason, all samples were reworked with a (2) jigsaw, (3) a belt sander with fine-grain sandpaper and (4) an air compressor.

A total of 24 pieces were made: 10x Model 1 error-free, 4x Models error error 1 and error 2 each, 1x special motive platypus and koi, or Models 4 and 5, and 4 samples without the brown coating (wood look).

2.2 Artificial Intelligence (AI) Tile Detection

In the project, image processing is carried out using Matlab. A webcam of the company CamPark of the type PC02 is used (see also Figure 1), and documentation in (Heiden et al., 2024).

The program workflow is shown in Figure 3. (1) First a function is created by the user to get inputs and outputs and, as in this case, these are reused in another file. With ‘Bildverarbeitung002.m’ (see the program code in (Heiden et al., 2024)), a single image is captured with the camera. The captured image is always an RGB image. This program uses the standard resolution of the camera. Image corrections are then made. A structuring element is a binary-valued neighbourhood, either 2-D or multi-dimensional, in which the true pixels are included in the morphological calculation and the false pixels are not. The central pixel

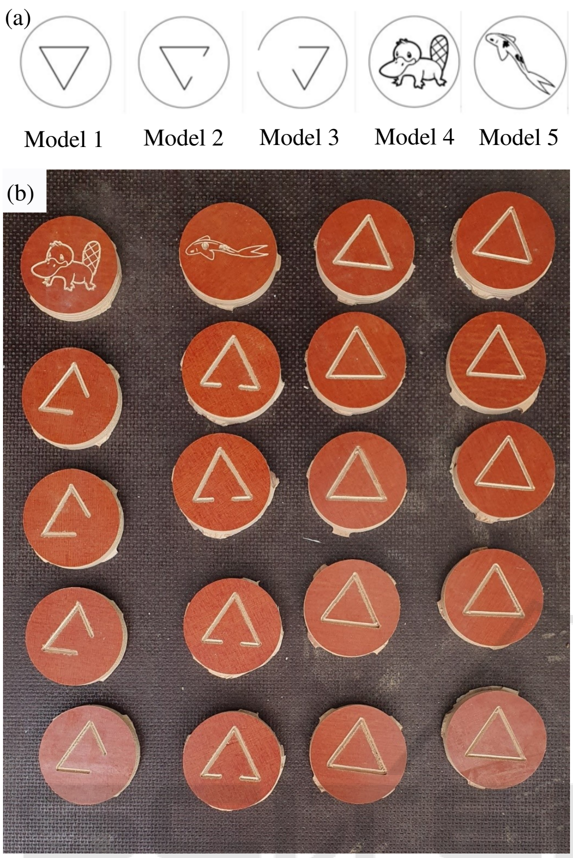

Figure 2: Tile Quality Test Parts - Constructed and Real Implementation.

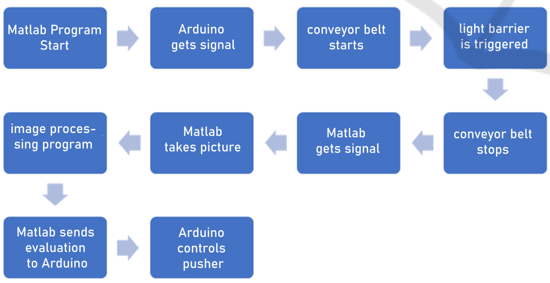

Figure 3: Program Workflow.

of the structuring element, called the origin, identifies the pixel in the image to be processed. In this case, 'strel("disk",r)' creates a disk-shaped structuring element, where r indicates the radius, here 15. Then, using this structuring element, 'imdilate' expands the image.

The next step is to count the black pixels, this is done to determine if the sample is a good or bad pattern. First, the greyscale image is binarised by thresholding, replacing all values above a globally defined threshold with ones and setting all other values to zeros. Since the image has now been divided into zeros and ones, the black pixels must be counted, i.e. all ones, since the zeros represent the white pixels.

(2) The file 'createMask1.m' (see the program code in (Heiden et al., 2024)) is used to differentiate those colour areas that occur on the pattern pane. The Color Thresholder app is used for this. It automatically generates a code. The Color Thresholder app can be used to segment colour images by threshold values for the colour channels based on different colour spaces. With this app, one can create a binary segmentation mask for a colour image. Color Thresholder supports segmentation in four colour spaces. In each colour space, the application displays the image, the three colour channels and the colour value of all pixels as points in a 3D colour space diagram. One can select the colours contained in the mask by displaying the colour channel values in a window or by drawing Regions of Interest (ROI) in the image or in the 3D colour space diagram. In the following we show how to segment an image and create a mask image using the Color Thresholder app. With the pattern slice, the pattern (triangle) is segmented from the background (slice and conveyor belt) based on colour values. The image can be segmented in several of the colour spaces supported by the app, as one colour space can isolate a particular colour better than another. In the project, the RGB colour space is chosen because it gave the best result. In each of the supported colour spaces, one can first perform an automatic segmentation by selecting an area in the foreground or background. Then one can refine the segmentation using the colour component controls provided by the application. Once the segmentation is complete, the results are saved, a mask image is created and the Matlab code used by the app to perform the segmentation is retrieved.

(3) Finally, the communication between Matlab and Arduino is implented. This is done by the program 'KommunikationMaltabArduino002.m' (see the program code in (Heiden et al., 2024)).

2.3 Construction of the IoT Device Demonstration Prototype

To demonstrate and evaluate the process of tile quality detection an IoT device is needed. Therefore, a prototype was produced by common manufacturing processes. In other words, a conveyor belt has been built, driven by a 12 VDC gear-motor to generate enough torque for the application. All needed work steps were achieved by in house manufacturing. These include some of the following manufacturing processes: several joining and cutting methods containing CNC cutting, 3D-Modelling, welding, soldering and other commonly known procedures. Moreover, all the electrical requirements needed to be addressed as well, whereas the implementation belongs to and has a huge impact on the proper functionality. To address this necessity up to date methods were used, like prototype circuit board cutting with an application related CNC cutting machine (details can be found in the project documentation in (Heiden et al., 2024)). Moreover, the choice of all the electrical components were chosen to allow for an interaction between all the used sensors and actuators. The prototype-like circuit board construction is needed, because the whole IoT-device is powered by an external 12VDC energy source, but some of the components need an individual DC potential to operate. An Arduino UNO was used to process all the sensor signals and operate the actuators, requiring a custom Arduino shield.

2.4 Control System of Prototype

The control system of the application is based upon the Arduino UNO microcontroller, thus the coding was ensued in the Arduino development environment. The inspection process can be portrayed and explained via the flow chart diagram in Figure 3 and 4.

It starts with a press of a physical button (there is an option to start the process through serial monitor as well). As a result, the DC motor powers on and puts the conveyor belt in motion. If the photoelectric sensor picks up a signal the belt is stopped in 1.5 seconds. This allows for the object to align directly under the camera, which is responsible for creating an image of the inspected object. While it is positioned under the camera, the Arduino transmits a signal via serial communication to the connected laptop that runs the Matlab software, where the image processing takes place. While the Matlab process takes place, the state of the device stays the same. Afterwards a signal is sent back, which consists either of a zero (‘0’) or one (‘1’). The Arduino therefore needs to decipher the signal and acts then according to the outcome. A ‘1’ signal causes to move the stepper motor to the left, where the qualified objects are sorted and the ‘0’ to the right. Finally, the conveyor belt starts moving again to successfully sort the inspected object.

The whole CNC system consists of the following components:

- DC Motor – for powering the conveyor belt
- Stepper motor – Stepper is attached with a custom-made sorter to angle the objects

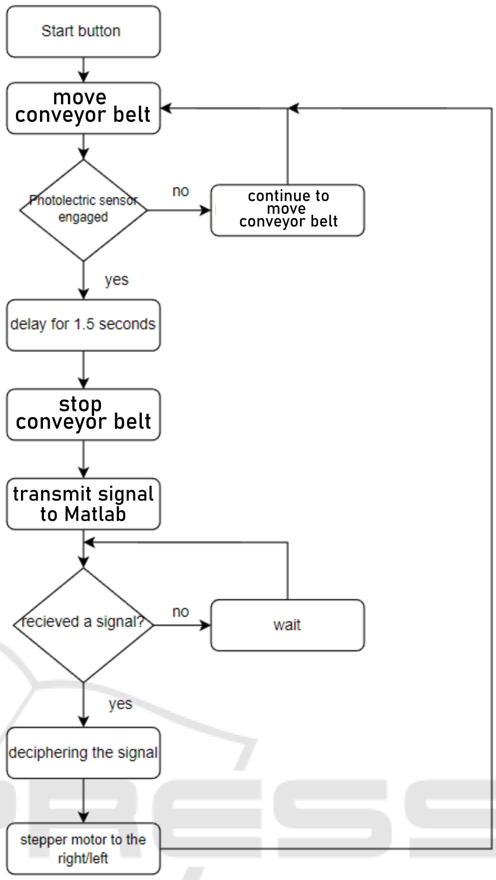

Figure 4: Flow Sheet Program of the Tile Quality Check Prototype.

- Photoelectric sensor – to sense the objects, which pass by
- Limit switch – mounted to the left and right of the stepper motor to limit its movement
- Start key – to power on the device

The Arduino code consists of switch cases, which allow the device to move from case to case accordingly in a loop. Moreover, multiple time variables are used, in order to keep track of the process and to control its movements as well as the integrated Arduino functions such as millis() for the time measurements and isAlpha() and isAlphanumeric() for deciphering the received signals. A thing to keep in mind, while using serial communication as indicators for the switch sequences, is to regularly clear the contents out of the serial monitor, due to the possibility of reading the older or wrong input.

3 DISCUSSION

The first test[1] has shown, that there is a huge dependency of the light condition on the effective detection. The sensitivity to this problem is also reported in industry in (Coskun et al., 2022), and counter measures have been taken by, e.g. illumination equipment. Although the general overall problem has been simplified by the round geometry of the tiles, there remain problems of pattern detection, although the mechanical handling works fine.

Another point is that in such experiments then the statistical results are automatically and not manually generated. By this different algorithms could be analysed systematically in an empirical fashion, which is now limited to a lot of manual interaction.

Concerning real world applications, this prototype is usable in classroom settings, as it is small, mobile and easy to transport. For a scale-up of this prototype to industrial applications, the Matlab automation can easily be used and adapted for industrial environments. The before mentioned problems with scanning quality, has additionally to be addressed.

The overall process is not fully automated, as there has to be an interaction of the operation with the tiles and the user who runs the Matlab program. For an industrial application procedure it would be necessary that the process works autonomously and reliable.

An automated testing extension for test data generation and to make automated testing and its statistical data evaluation of the quality of the testing would improve the current prototype system significantly, which would then be in the upscaled version a kind of auto-calibration function in the field.

4 CONCLUSION AND OUTLOOK

An IoT prototype for detection patterns in tiles, that are related to quality issues in production is investigated in this paper. For this an IoT device has been designed, built, and tested. The result is the IoT-prototype shown in Figure 1. The AI is implemented with a Matlab program. The communication is triggered by Matlab and the main control of the device on the other side is done by an Arduino UNO program, powering the hardware of the given IoT device.

The future outlook of the work is to improve the light sensitivity of the device, the robustness, of the detections algorithms with regard to different tiles and the implementation of more sophisticated AI models like, e.g. deep learning. Another point is to further generalise (a) the prototype, so that it can be used for the detection of more complex patterns, and (b) patterns that are derived from real examples from industry. In those cases also the velocity is important, which then means that a different than purely Matlab and Arduino based method for this application could possibly be taken into consideration for improving the effective productivity, reliability, reproducibility, effective detection efficiency, etc.

Finally, it has to be mentioned that the solution of the problem of pattern detection is only one aspect of quality issues. The other is of improving the process, which then means that the patterns are subject to an evolutionary change in the course of continuous process improvement. This then also will open up the question how to implement this pattern changing situation into the pattern detection problem on the one hand and to use AI to avoid quality issues by process identification and process parameter improvement on the other side before those quality issues occur, which then also means that the 'quality type' or the meta-quality of detection could change in the course of this continuous improvement process.

This, as well could then be implemented in a future new version of an educational prototype, which can then be understood as a downscaled educational IoT twin of the industrial original IoT device.

[1] See project documentation in (Heiden et al., 2024).

REFERENCES

Borchers, J. (2013). *Arduino in a Nutshell*. http://hci.rwth-aachen.de/arduino

Choo, K.-K. R., Bishop, M., Glisson, W., and Nance, K. (2018). Internet- and cloud-of-things cybersecurity research challenges and advances. 74:275–276.

Corallo, A., Lazoi, M., Lezzi, M., and Luperto, A. (2022). Cybersecurity awareness in the context of the industrial internet of things: A systematic literature review. 137:103614.

Coskun, H., Yigit, T., and Üncü, I. S. (2022). Integration of digital quality control for intelligent manufacturing of industrial ceramic tiles. 48(23):34210–34233.

Dong, G., Sun, S., Wu, N., Chen, X., Huang, P., and Wang, Z. (2022). A rapid detection method for the surface defects of mosaic ceramic tiles. 48(11):15462–15469.

Frei, F. (2017). Production optimisation based on a case study at the company CerMed inc., original in German: Produktionsoptimierung anhand einer Fallstudie bei der Firma CerMed GmbH. Bachelor Thesis, Villach, Carinthia University of Applied Sciences.

Gershenfeld, N. (2006). Unleash your creativity in a fab lab. Feb. 2006, video, length: 17'15". https://www.ted.com/talks/neil_gershenfeld_on_fab_labs

Gibson, I., Rosen, D., and Stucker, B. (2015). *Additive Manufacturing Technologies 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. Springer.

Hanzaei, S. H., Afshar, A., and Barazandeh, F. (2017). Automatic detection and classification of the ceramic tiles' surface defects. 66:174–189.

Heiden, B., Pacnik, D., Pregl, J., Müller, M., and Leitner, S. (2024). Source code for the tile quality detection device. https://github.com/BernhardHeiden/TileQualityDetectionDevice

Kief, H. (2015). *CNC-Handbuch CNC, DNC, CAD, CAM, FFS, SPS, RPD, LAN, CNC-Maschinen, CNC-Roboter, Antriebe, Energieeffizienz, Werkzeuge, Industrie 4.0, Fertigungstechnik, Richtlinien, Normen, Simulation, Fachwortverzeichnis*. Hanser, München.

Lemoine, F., Aubonnet, T., and Simoni, N. (2020). Self-assemble-featured internet of things. 112:41–57.

Lindström, J., Eliasson, J., Hermansson, A., Blomstedt, F., and Kyösti, P. (2018). Cybersecurity level in IPS 2 : A case study of two industrial internet-based SME offerings. 73:222–227.

Mesch, F. (1985). Impact of new technology on measurement education. 3(1):44–48.

Wan, G., Fang, H., Wang, D., Yan, J., and Xie, B. (2022). Ceramic tile surface defect detection based on deep learning. 48(8):11085–11093.

Zhang, H., Peng, L., and Lei, G. (2022). Saliency detection for surface defects of ceramic tile. 48(21):32113–32124.