UNIVERSITY OF TURKU

Part detection with 2D machine vision utilizing traditional methods

Physical and algorithmic foundations of machine vision systems

Mechanical Engineering/Department of Mechanical and Materials Engineering

Bachelor's thesis

Author:
Veeti Suominen

15.5.2026
Turku

Bachelor's thesis

Subject: Mechanical Engineering

Author(s): Veeti Suominen

Title: Part detection with a 2D smart camera utilizing traditional methods; Physical and algorithmic foundations of machine vision systems

Supervisor(s): Jarno Salminen, Wallace Bessa

Number of pages: 36 pages

Date: 15.5.2026

Abstract: This study investigates the physical and algorithmic foundations of 2D machine vision systems in part detection. The study begins with a literature review outlining the core components of a machine vision system, such as lighting, optics, and image processing, as well as exploring its industrial applications and benefits, while contrasting traditional rule-based algorithms with modern deep learning-based methods.

The primary purpose of the research is to demonstrate the capabilities and limitations of 2D machine vision utilizing traditional algorithms when exposed to real-world complexities rather than ideal conditions and objects. To achieve this, an experimental study was conducted using a 2D smart camera and a traditional rule-based machine vision tool. Visual and coordinate data were collected from practical test scenarios involving reflective surfaces, translucent materials, low-contrast environments, and varying object heights to demonstrate the boundaries of 2D perspective and traditional edge detection.

The findings demonstrate that traditional 2D systems depend strictly on clear optical gradients and that their lack of depth perception causes perspective distortions when object placement varies. In conclusion, while traditional 2D vision is highly efficient in controlled low-level settings, overcoming physical constraints in dynamic and complex manufacturing scenes requires considering more advanced technologies like deep learning-based methods and 3D vision systems.

Keywords: machine vision, smart camera, computer vision, part detection, object detection, edge detection, traditional machine vision, machine learning, learning-based machine vision, 2D machine vision

Use of AI -tools: Keenious was used for finding sources. Google Gemini was used for image generation, proofreading and grammatical correction.

Table of contents

1 Introduction

1.1 Background on machine vision

Traditional automation is built on rigid pre-programmed robot sequences and inspection and assembly tasks done by human workers. In modern automation there is a strong desire to transition to flexible and adaptable automated operations, and this transition relies heavily on the ability of machines to perceive their environment. Machine vision (MV) is the primary field and technology that enables this transition. By allowing robots and machines to “see”, MV -systems can eliminate deficiencies by being able to identify, inspect, locate and manipulate objects, integrating to changes in conditions and continuously narrowing the gap between digital algorithms and the unpredictable physical world. (Smith et al., 2021)

Computer vision is a field of artificial intelligence (AI), focusing on image understanding algorithms and technologies (Szeliski, 2011). Machine vision is a sub-branch of computer vision and can be seen as the practical realization of computer vision techniques, focusing specifically on practical industrial applications that involve a significant visual component. The desire to use MV in industrial processes is often motivated by goals to reduce cost by increasing efficiency and productivity, reducing error and improving quality or to gather data. MV can also substitute for an absence of available skilled labour or replace human workers in tasks that could potentially be harmful, highly fatiguing or dangerous. With developments constantly being made in coupling machine learning and robots with MV, the technology offers capabilities even closer to that of human vision and -operation. (Smith et al., 2021)

1.2 Research problem and objectives

In this study, we will focus on 2D machine vision, even though many aspects are shared between 3D MV -systems as well. The aim for the literature review is to give an overview of the fundamental components in a machine vision system regarding both physical and processing elements, and to showcase the different industrial applications and benefits of machine vision. The primary objective of the experimental study is to evaluate the capabilities and limitations of a 2D smart camera machine vision system in part

detection across a variety of parts with certain features and in other physical and optical challenges. Rather than focusing on ideal conditions, the study's aim is to showcase real-world complexities and scenarios, such as highly reflective surfaces, low contrast environments and translucent materials. The kind of problems that an operator setting up an industrial MV application might run into. The study showcases problems while using traditional detection algorithms and how these problems manifest themselves in software. It is also shown that how some of these challenges can be combated with basic-level machine vision tools.

1.3 Structure of the thesis

The thesis structure progresses from a literature review on MV to practical experiments and problem-solving: The second chapter is a literature review on machine vision, brief history, developments, core components, image processing and algorithms, and industrial applications and benefits. The third chapter introduces the equipment and research methodology. The fourth chapter presents the experimental results, and the two final chapters provide discussion and analysis of the results and conclude the thesis.

2 Machine vision foundations – literature review

2.1 Brief history and evolution of machine vision

The field of machine vision has evolved significantly over many decades. The field has transitioned from very early artificial intelligence ambitions to highly mathematical and data-driven disciplines. Computer vision originated in the 1960s. It originated as the visual perception component of an ambitious agenda to mimic human intelligence. At the time, it was believed to be an easy step along the path to solving more difficult problems such as high-level reasoning, but it proved to be slightly more difficult than that (Szeliski, 2011). In the 1970s, an MIT -engineer developed an approach for scene understanding trough computer vision, and this marks the beginning of real-world object tracking and low-level vision tasks such as detection and segmentation (Tzampazaki et al., 2024). Image segmentation is a core activity of machine vision, and it is the process of partitioning an image into distinct segments through labelling every pixel, allowing for similar pixels to be grouped together or for contours to be found.

In the 1980s, the development of smart cameras and optical character recognition (OCR) systems widened the possible applications of machine vision in industry, but the technology remained expensive and needed professional operators to utilize. OCR is a machine vision system or feature that can read characters and text, widely used in manufacturing-related barcode and label reading tasks. In the 1990s machine vision started to rapidly invade the industrial environment. This was due to technological advancements in computers, AI, image sensors, control architectures and automation as a whole field. Companies started to sell commercial machine vision systems, lowering the acquisition costs and making MV -systems more accessible to everyone. This accessibility simultaneously enhanced the possibilities for machine vision to further advance as a field. (Tzampazaki et al., 2024)

Since the early 1990s, machine vision has been developing rapidly. Nowadays, machine vision is used for a multitude of problems in the industry, such as defection, classification, identification, material quality, control, monitoring, maintenance, storage etc. (Tzampazaki et al., 2024). The more recent emphasized focus on conjoining MV and machine learning has been made possible by transformative developments in

the field of AI and has allowed to move machine vision capabilities even closer to that of human vision, even surpassing it in many areas and applications. These recent advances are finally starting to answer that ambitious agenda that originated in the 1970's (Smith et al., 2021).

2.2 Core components of a machine vision system

2.2.1 Lighting

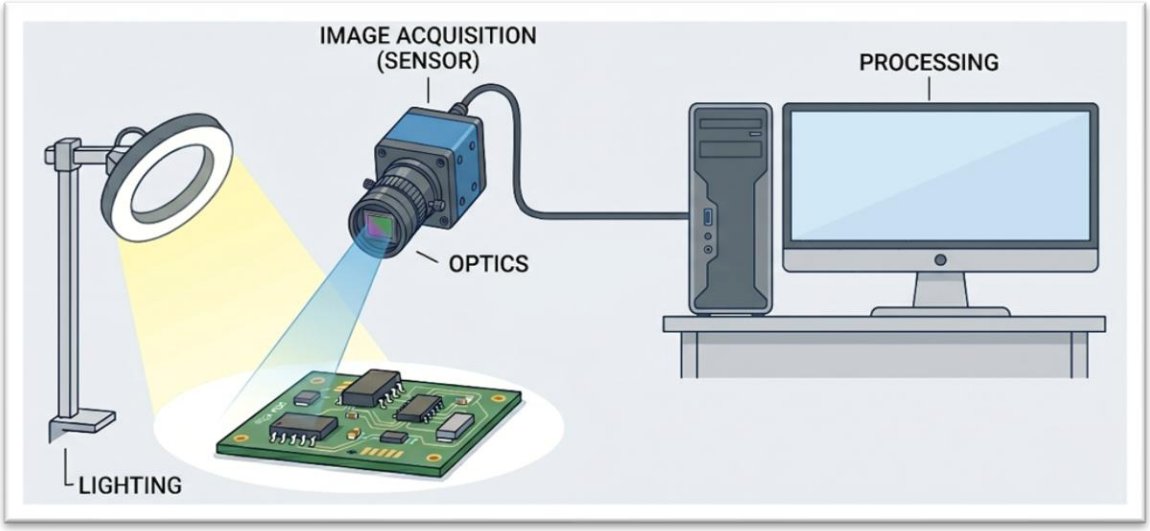

Figure 1. Machine vision system components represented in an example scenario with an PCB. Image generated by Google Gemini (Nano Banana 2), 2026.

Lighting or illumination is a critical component in any MV setup. The lightning element is labelled in Figure 1. along with other core elements of an MV system. In a sense, all information processed originally comes from the light (Hornberg, 2017). Machine vision algorithms prefer consistent lighting conditions that provide uniform image quality to function as reliably as possible. Environments with challenging lighting conditions remain a significant obstacle for MV solutions, although advances in sensor technology have facilitated the acquisition of more complex data (Kim et al., 2026). The main objective of sufficient lighting is to highlight the features of an object that needs to be inspected and to minimize noise, such as shadows and unpredictable reflections. MV applications are almost always fitted with an active lighting solution, meaning there is an arbitrary light source illuminating the object and therefore the choice of light source is an important factor (Golnabi and Asadpour, 2007). There are a number of illumination

technologies such as fiber optics, halogen lamps, light-emitting diodes, and these can utilize numerous techniques, such as ring-, spot-, back- and diffuse lighting (Golnabi and Asadpour, 2007). As mentioned before, all components of a MV system must work together seamlessly, and therefore the choice of optics also influences the type of lighting required, effecting factors such as light wavelength, light colour, light intensity etc.

2.2.2 Optics

Once the object is properly illuminated, the light from the scene must still pass through the optics before reaching the camera sensor (Szeliski, 2011). This thesis will focus only on optics aspects relevant to MV. One of the main factors depicting optics choices is the amount of area you want to capture in the camera view. The lens changes the field of view of the camera, which essentially is a certain amount in degrees, that the camera sees. For detecting an object in a confidently predetermined place, a lower FOV (field-of-view) will be suitable, but for broader and more unpredictable scenarios, a wider FOV covers a potentially larger area. Depth of field is the distance away from the camera that stays in sharp focus, also affected by the optics choices and adjustments. Choosing correct optics is crucial to produce a high-quality image of the object from a desired distance. Certain type of lenses or optics can also reduce the amount of perspective retained in an image to flatten out depth out of objects, which can be beneficial in some applications, where depth information is disadvantageous. The optics are used to focus and retrieve the light information wanted, but the image sensor captures the light in a form that can be digitized.

2.2.3 Image acquisition

The image acquisition is an process where an image sensor is used to create an optical image that can be converted into a digital image (Golnabi and Asadpour, 2007). There are various types of sensors, differing from each other in features and underlying technology. Resolution is the number of pixels the sensor captures, directly impacting the level of detail in an image. The shutter type, either global- or rolling stutter, depicts the way the pixels are captured. A global shutter captures all the pixels simultaneously, good for imaging fast-moving objects, and a rolling shutter captures the pixels

sequentially, better for slow-moving objects. The sensor shutter speed tells how many frames per second the sensor can capture, crucial for high-speed object detection applications, such as fast production lines. The sensitivity depicts the sensor's ability to perform in low light scenarios. All these before mentioned features can be considered for different kinds of applications. The most commonly used sensors are type CMOS (Complementary Metal Oxide Semiconductor) -sensors due to widely applicable features and performance, and as well due to high-speed readout, low power consumption and cost-effectiveness (Hornberg, 2017).

2.2.4 Integration of components

In conclusion, the effectiveness of a machine vision system depends on the proper integration of all its core components. While processing has not yet been discussed, it is clear that the quality of the raw input data, dictated by lighting, optics and the sensor, directly affects the challenges faced in the software side of things. The physical hardware serves as the primary method to control environmental variables. If the physical setup cannot create sufficient contrast or prevent reflections, the captured image will be of lower quality. A term often used in MV to present this trade-off between the quality of the physical setup, and the required digital processing is "Garbage in, Garbage out". The term is also used in other fields of machine learning and data science. It means that it is hard to produce reliable and efficient object detection applications if the input data is low quality and inconsistent.

Fortunately, advancements in machine vision technology have simplified the process of designing a machine vision system, especially for lower-level applications. Equipment such as smart cameras ease the process of designing a MV application by eliminating the need to consider and define every component. A smart camera is a device that has the computational power built into the housing, and they often ship with software. These ready-made setups and devices are built of components that have already been found to work well with each other and therefore make a functional MV system a bit more accessible and achievable for the consumer. Having established what components are needed and how they affect the quality of the input data, the next

section will cover the core processing algorithms used to analyse this data and perform reliable object detection.

2.3 Object detection

2.3.1 Image segmentation methods

Object detection algorithms are used to extract features and objects from images. The first step of object detection is image segmentation. Image segmentation is a process that seeks to partition an image into meaningful regions (Golnabi and Asadpour, 2007). There is no general theory for image segmentation, but various theories and methods (Yuheng and Hao, 2017). For simplicity, these various methods will be roughly represented by two main categories: traditional rule-based methods and more modern deep learning-based methods.

There are various traditional image segmentation methods, but threshold segmentation is the simplest and therefore beneficial to be used as an example. The algorithm directly divides the image grayscale information to background and foreground by a single threshold. The threshold is a set value somewhere in the grayscale or pixel intensity range of the image. The threshold can be set manually but is often done automatically by software. The simplified steps of threshold segmentation are represented on Figure 2. Essentially the steps are turning the image into grayscale, choosing a threshold and resulting in a black and white picture. The example shown in Figure 2. is an ideal situation, where the X-shape target has high contrast to the background, which makes finding a suitable threshold easy. After image segmentation, the next steps would be feature extraction and interpretation for an object detection result. The advantage of the threshold method is that the calculation is simple, fast and requires little computational power. The segmentation effect is achievable for situations where the target and background have high contrast. The disadvantage is that it is difficult to achieve good results for image segmentation problems where there is no significant grayscale difference. Since the method is purely relying on grayscale information, and is sensitive to noise and grayscale unevenness, it is often combined with other methods. (Yuheng and Hao, 2017)

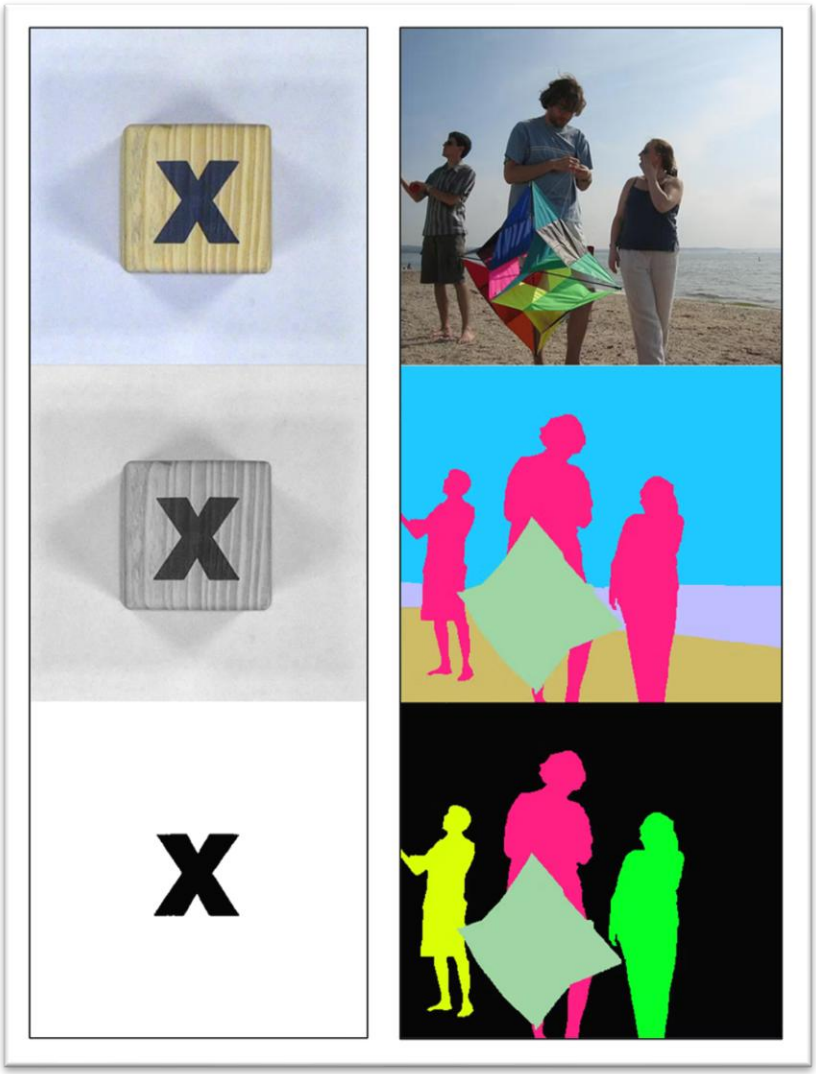

Figure 2. Left side shows the simplified steps of threshold image segmentation. The right side shows the simplifies steps of semantic- and instance image segmentation. Left image: Veeti Suominen, 2026. Right image: edited from https://encord.com/blog/guide-to-semantic-segmentation/, Nikolaj Buhl, 2024.

The second category of image segmentation methods is deep learning (DL) -based methods. This category further divides into different approaches, but the general basis is on utilizing neural network architectures, most notably CNNs (Convolutional-Neural-Network), which are trained on large datasets to perform image segmentation tasks (Smith et al., 2021). For machine vision applications the most notable advantage over traditional methods is the ability to remain accuracy and robustness with diverse objects, complex backgrounds and non-static conditions, such as varying lighting or unexpectable object positions and shapes (Sun et al., 2024). Overall, DL methods widen the scope of applications of MV significantly. In some ways, developing a deep learning-based solution to a machine vision task is simplified, compared to traditional methods,

since segmentation, feature extraction and interpretation steps are all combined into one stage. Deep-learning avoids the need to design and hand-code these steps, but instead the developer is required to possess a differing set of skills regarding implementing deep learning network architectures and tuning them. (Smith et al., 2021)

DL methods also have some problems regarding industrial deployment for a certain application. There is potentially a large amount of work required with training the AI model. Most applications utilize supervised learning AI models with large quantities of data. This raises a need to label (i.e. say what the image contains) the data, usually a manual process, which can take a lot of time, compared to the traditional image segmentation method, which requires no labelling. The labelling is used to identify the object classes in the present in the image for training the model. Another problem with deep learning is that often there may only be limited data available for the desired task. Both issues can to some extent be addressed using data augmentation. This is where a limited labelled dataset is altered with a range of automated image transformations to produce a larger artificial dataset, requiring less initial work. The trade-off is that the use of these methods is not expected to result in as good a performance as training using an authentic larger dataset. (Smith et al., 2021) Pre-trained models are also a growing trend, since they can largely reduce the amount of training and labels, and can be quite applicable to some industry tasks (Sun et al., 2024).

Deep learning-based methods allow for semantic segmentation of an image. This means subtracting more semantic and meaningful information of the image, rather than only edges or features. Semantic segmentation semantically annotates the different areas of the picture. The simplified steps of semantic segmentation are represented on the right side of Figure 2. In this approach, the image is inspected pixel by pixel and every pixel is annotated depending on predefined classes that the AI model has been trained to recognize, such as human, beach, water etc. After the initial semantic segmentation, the image consists of differently annotated areas. The last step is usually instance segmentation, which uses different area detection methods to differentiate these areas to their own unique instances, for example different humans. Semantic segmentation is used in machine vision applications such as defect detection, monitoring tool wear and sorting tasks (Smith et al., 2021).

2.3.2 Edge and object contour detection methods

The step following image segmentation is object recognition, Object recognition is essential for an actual beneficial result for a machine vision task. For object recognition there are various techniques and approaches. The type of technique available for use is significantly determined by the earlier steps in the process of object recognition, such as whether traditional rule-based or deep learning-based image segmentation methods were used. Image segmentation can be seen as the dual problem of object recognition, since they affect each other in terms of available methods. We can roughly divide the object recognition methods to traditional methods and learning-based methods. (Yang et al., 2022)

Traditional methods include local pattern-, edge grouping- and active contour methods. These methods all have relatively simple steps. They either apply operators directly to detect edges, or compute and minimize a cost or energy function to find contours. Usually, post-processing is done to link found edges or contours. These methods do not need data to learn, but their performance is poor compared to learning-based methods. Local pattern methods rely on filters, like ones used in image processing. The method is simple, fast, and widely used, but not so powerful for reliable edge detection. They are sensitive to noise, and do not distinguish real edges from texture regions, so they often result in discontinuous, chaotic edges rather than smooth closure contours. Local pattern method filters are often used as a pre-step for edge grouping. (Yang et al., 2022)

Edge grouping methods are better at detecting continuous and smooth contours. The method utilizes computing and minimizing cost functions and edge linking strategies to form continuous contours. The most used principles in edge grouping methods are evaluating proximity and continuity. The calculation of these evaluations is based on chosen principles, such as Gestalt principles. The evaluations are used to link edges that likely belong on the same contour to form a continuous contour. (Yang et al., 2022)

Active contour models are variational models for more precisely detecting and locating object contours. Unlike edge grouping, which links discrete pixels, active contour

models utilize an iterative function process to evolve into a shape representing the object contours. The function, called the energy function, balances two competing forces, the internal and external force. The internal force maintains the continuity and smoothness of the curve, while the external force pulls the curve towards the image features, such as gradients or edges, resulting in a curve resembling the object shape through many iterations. (Yang et al., 2022)

The second major category of object recognition techniques is learning-based methods, which learn to classify or locate edges based on extracted visual features rather than mathematical rules. These are primarily divided into classical learning- and deep learning-based methods. Classical learning-based methods rely on manually engineered features, such as brightness gradients or specific edge patterns, extracted either pixel-by-pixel or in small patches. These features are used to train standard classifiers, like support vector machines or random forests, to detect boundaries. While being effective at filtering out background noise, designing optimal features is difficult and the multi-step classification process can be too computationally heavy for real-time applications. (Yang et al., 2022)

Advancing on classical methods, deep learning-based methods have become the modern standard. Instead of relying on manual feature engineering, deep convolutional neural networks (CNNs) automatically learn the most important visual patterns directly from raw image data. Deep learning methods excel due to their automated feature extraction capabilities. However, they require large amounts of labelled training data and significant computational power, similar to the deep-learning based image segmentation methods. Furthermore, unlike traditional approaches where the mathematical logic is visible, deep neural networks essentially function as "black boxes," meaning that they lack the readability to easily explain how the model found a specific contour. (Yang et al., 2022)

2.4 Industrial applications and benefits

The last chapter of the literature review servers as a bridge between the practical applications of machine vision and the discussed algorithmic foundations and physical components. The deployment of these algorithms in real-world industrial applications

directly translate to automated systems capable of performing repetitive, precise and complex vision tasks at speeds exceeding human capabilities. First, we will look at the applications and benefits with machine vision utilizing traditional methods.

In manufacturing, traditional machine vision is heavily utilized for automated visual inspection (AVI), process control, parts identification and robotic guidance. Typical tasks include gauging object dimensions, verifying assembly alignments, sorting products and providing position data for robotic movements, such as robot welding or picking operations. The primary advantage achieved with traditional systems over human workers is cost-effectiveness. For example, by replacing manual quality inspection, factories can significantly reduce labour costs and accomplish inspection of every product in mass production environments. Utilizing machine vision can also eliminate errors caused by factors such as worker fatigue, and therefore help ensure a higher baseline of product reliability and safety. (Golnabi and Asadpour, 2007)

Traditional machine vision isn't however suitable for all environments. From a production view, traditional systems are highly rigid. They require strictly controlled physical environments and don't adapt well to complex conditions, such as fluctuating lighting or unexpectedly altered parts. Traditional systems typically fail to adapt to variations without manual mechanical and programming adjustments. (Zhang et al., 2021)

With the shift towards industry 5.0 aiming for machine and human resources to work together, learning-based machine vision is developed to handle unstructured and dynamic industrial conditions (Tzampazaki et al., 2024). The range of applications is already vast, but it is also extendable through development, since there are no similar algorithmic limits as with traditional machine vision. Applications for learning-based MV include tasks such as detecting unpredictable surface anomalies, active dimension measurement, many varieties of part inspection, and real-time advanced object localization for robotic arms (Wang, 2022). In a modern setting with a collaborative robot, visions systems are also used to monitor human workers and predict their movements allowing for robots to safely assist in shared tasks (Tzampazaki et al., 2024). Learning-based systems allow for true flexible manufacturing. They can effortlessly

adapt to diverse product shapes and more complex industrial backgrounds without the need for physical reconfiguration (Sun et al., 2024). When incorporated with the IioT (Industrial internet of things), these systems have the capability to act as intelligent nodes that provide factory floor data in real-time, aiding in processes such as autonomous decision-making and predictive maintenance (Tzampazaki et al., 2024).

The primary challenges with utilizing learning-based MV systems are deployment time, data acquisition and therefore cost. Training these systems require large volumes of accurately labelled data specific to the targets at hand, which is a manual and costly process. Also setting up a learning-based system requires more specialized knowledge and skillset compared to traditional MV. As mentioned before, deep learning-based networks operate as “black boxes”, meaning that in an industrial setting, if an error occurs on the production line, it lacks the readability to explain how the AI model arrived at a specific decision, complicating troubleshooting and quality assurance.

3 Experimental setup

3.1 Test environment and equipment

The experiments are conducted at Vertek Oy's office in Uusikaupunki, using their equipment. The equipment was preassembled, setup and tested, ready to be used.

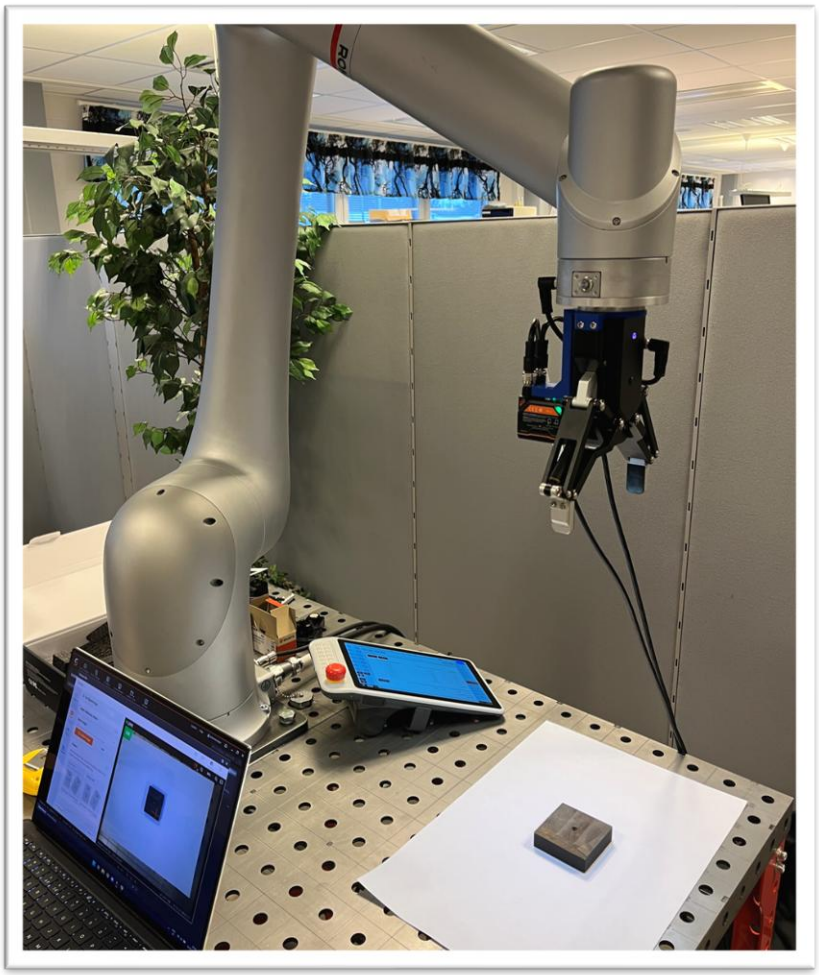

Figure 3. Overview of the experimental setup equipment. Picture: Veeti Suominen, 2026.

The equipment is shown in Figure 3. Relevant equipment for our experiments shown in the picture are: Rokae CR12-C -robotic arm and its controller, the Hikrobot MV-SC3030XC-08M-WBN -2D smart camera, and a laptop for operating the software. Specifications for the Rokae won't be listed, since it is mostly used just as a platform for mounting the camera and therefore is not relevant.

The Hikrobot camera is an integrated smart camera architecture. This means that the camera features built-in hardware processing and embedded vision algorithms. This

allows for the camera to process and send results directly to the robot or PLC without requiring a separate processing PC. In our case, the separate PC is used for setting up the different scenarios and tasks, and monitoring results. The camera features a 3-megapixel colour CMOS -sensor with a global shutter, outputting an image resolution of 2048 x 1536 pixels. The camera has a focal length of 8 mm, which paired with the resolution results in a single pixel accuracy of 0.011 mm at 25 mm working distance, and 0.863 mm at 2000 mm working distance. Working distance represents the distance between the target plane and the camera lens. The camera has a framerate of 40 fps. The camera has 3 modes of active lighting around the lens for illuminating the target object, and the lens is polarized and diffused to help control environmental lighting. All the mentioned specs were taken from a data sheet available on Hikrobot's website on 12.5.2026. The smart camera is mounted at the tool end of the robotic arm, as seen in Figure 3. The setup includes a gripper system, but it is not used in our testing. The gripper cannot be taken off, since the camera mount is designed around it, but it fortunately doesn't affect the experiments.

The software is Hikrobot's SCMVS (Smart Camera Machine Vision Software), which is shipped with the smart camera and intended for client use. Software will be discussed further later.

The physical arrangement for most experiments can be seen in Figure 3., with relevant factors being that the camera is mounted vertically, facing perpendicular to the table surface, and that the camera height and position is fixed to the shown position. For illuminating the objects, the office space has overhead ceiling lights that are turned on along with utilizing the smart camera's integrated active lighting.

3.2 Methodology

3.2.1 Calibration

With the camera height being fixed, a pixel-to-millimeter calibration can be done, which helps to showcase the results in a more informative unit. The calibration is done using a "checkerboard" pattern, where the size of a single square is known. The calibration window inside the software is shown in Figure 4. In our case, the square height on our

printed pattern is 30 mm. The software finds the corners of the squares and uses the information to form a matrix, that transforms pixel values to millimeters and accounts for lens distortion. Lens distortion is an effect that causes curvature at the outer edges of the view and can distort outputted position values. The camera height needs to be fixed, since altering it changes the calibration parameters, and recalibration would be necessary. It is to be noted that since the camera is 2D, all the positional values extracted will be on the plane that the calibration was done on, in our case the table surface.

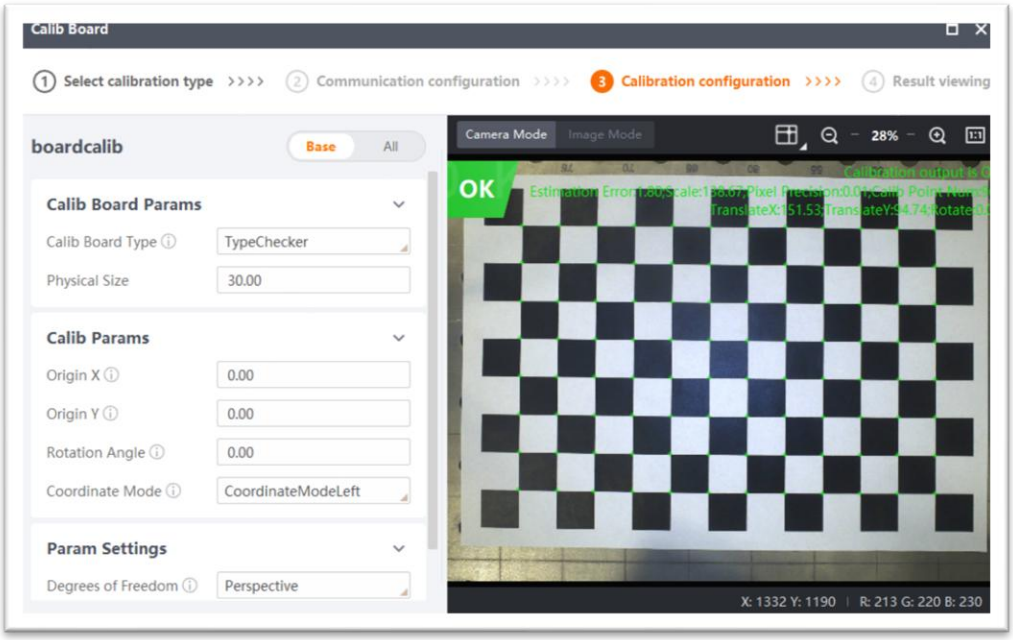

Figure 4. Pixel-to-millimeter calibration window. Screenshot from HikroBot SCMVS -software, 2026.

3.2.2 Default procedure

Here the default setup or procedure for the experiments is described. This is done to achieve more reliable and comparable results.

An automatic calibration of camera values is done with every new experiment or object. This happens through “Auto Brightness” and “Auto Focus” in the software, and it essentially automatically optimizes camera parameters to achieve a clear and focused image.

For every experiment, the same ROI (Range-of-interest) is used. The ROI defines the area of the image that the software considers when detecting objects. It is defined to be the full plain white background that we used.

Results will be mostly displayed in the feature template teaching window of the Contour Count -tool, to showcase found contours. This is further explained in the next section.

3.2.3 Contour count -tool

The tool used for every experiment is the “Contour Count” -tool inside the SCMVS -software. The function of the tool is to determine the presence of contours through contour matching and to count the number of found matching contours. The tool uses traditional rule-based edge detection algorithms relying on image processing to find edges, and then forms connected, closed or nearly closed contours through linking edges. The teaching of the feature template is done on a single-image basis. The tool has capabilities to utilize embedded deep-learning algorithms to deal with more complex scenes and strengthen recognition, allowing the tool to identify objects even when contours are partially hidden, for example with overlapping objects. For our use cases, the deep-learning capabilities won’t be beneficial since our experiment scenes do not present the type of problems that the embedded AI could be used for.

The first step with using the tool is to take a base image of the target object in the expected conditions and background. This base image is used to build the feature template, that will later be used to recognize the target object through template matching. An example of the view of building the feature template is shown on Figure 5.

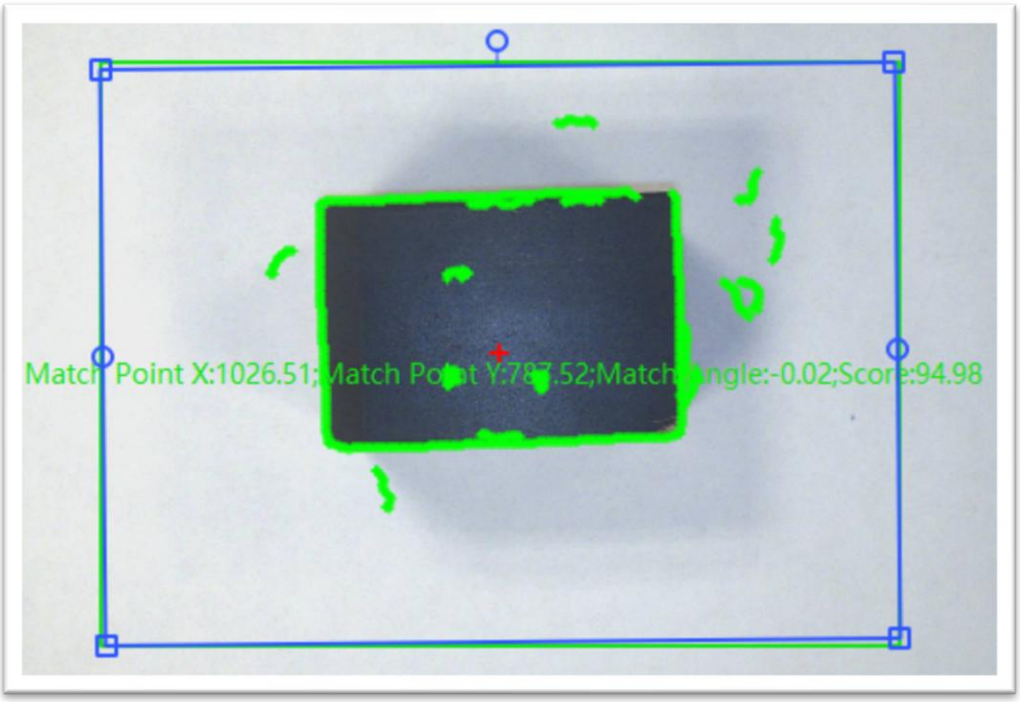

Figure 5. An example of a Contour Count -tool feature template building view. Screenshot from Hikrobot SCMVS -software, Veeti Suominen, 2026.

The blue rectangle is the selected detection area. The detection area defines the area of the base image that is be processed to find contours. The found contours are represented by the green lines. In the example picture contours are confidently found around the whole black object, since there is high contrast between the object and the background. Unwanted contours can be “erased” from the feature template by drawing red “shield” areas over them. These are represented as areas with a red border and hue, shown in later pictures.

There are also a few parameters that can be adjusted to enhance initial contour detection as well as object recognition while running the tool. The main parameters that will be adjusted are:

Camera parameters: These were automatically set, but exposure and gamma can be adjusted manually for better results.

Grayscale threshold: This can be set to Auto or Manual. On the manual mode, this parameter sets the grayscale contrast threshold required to define an edge. Increasing the value filters out faint or low-contrast edges and can be used to eliminate background noise or soft shadows.

Match polarity: This dictates whether the algorithm considers the direction of the grayscale gradient (e.g. light object on dark background and vice versa) to detect edges. If set to “Considered” the algorithm will only detect contours with the matching direction of gradient as the learned feature template. This shortens the finding time.

Minimum score: This determines the confidence threshold for recognition. The software gives out a matching score (0-100), which indicates how well the found contours match the feature template. A high minimum score requires the detected contours to perfectly mimic the feature template, reducing false positives but simultaneously demanding highly consistent object presentation. Lowering the value allows for a bit more flexibility in the recognition but risks false positives, especially in more complex scenes with multiple objects.

Scale- and Angle range: These parameters define the amount of angular- or size scale offset the object recognition searches and allows for. For example, a manual scale mode range of 90%-110% allows for recognition of matching contours that are 10% larger or smaller than the learned feature template. This can allow for situations where the target object can be closer or farther away from the camera, making it seem smaller or larger. Angle range works in the same way comparing the found objects boundary box angle to the learned template. The range is set in degrees from -180 to 180.

4 Experiments and results

4.1 Material related challenges

This section includes the experiments done regarding the challenges that different materials can cause for traditional machine vision. First target object is a small shiny pot.

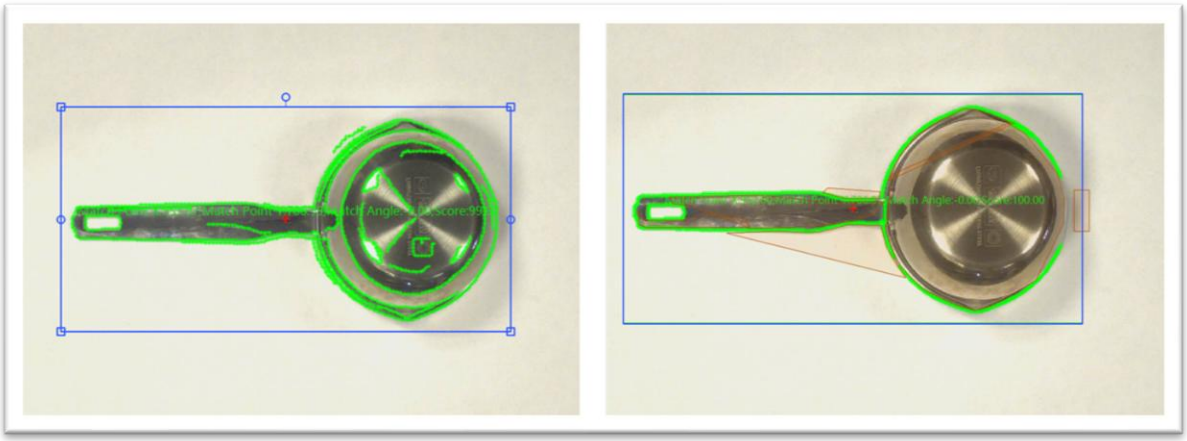

Figure 6. Experiment with small shiny pot. On the left is the initial contour recognition, and on the right is the functional feature template. Screenshot from Hikrobot SCMVS -software, Veeti Suominen, 2026.

The significant challenge in this experiment is reflective surfaces. As seen on the left side of Figure 6. the software tries to draw a contour around the object. But because the metal reflects light unevenly, the software interprets the bright reflections as background in the object, causing “ghost” edges. If you try to correct this only via adjusting the grayscale threshold, you often end up erasing the actual edges of the object as well. The traditional algorithm is somewhat powerless against reflective objects since the optical data can be very chaotic. The detection is made functional through software methods by shielding out the unwanted ghost edges and adjusting the grayscale threshold. These shielded areas and the working feature template can be seen on the right side of Figure 6. What is left after shielding is the outer contour and a small hole on the handle of the target object, and these can be easily detected from the white background. Reflective materials can also unexpectedly cause specular reflections straight at the lens, causing a huge white spot in the image and ruining all chances of detection. These are often avoided by moving the camera a bit to the side, while still having the light source above the object.

The second experiment regarding object material has to do with translucent materials. The chosen target object is a translucent plastic bottle.

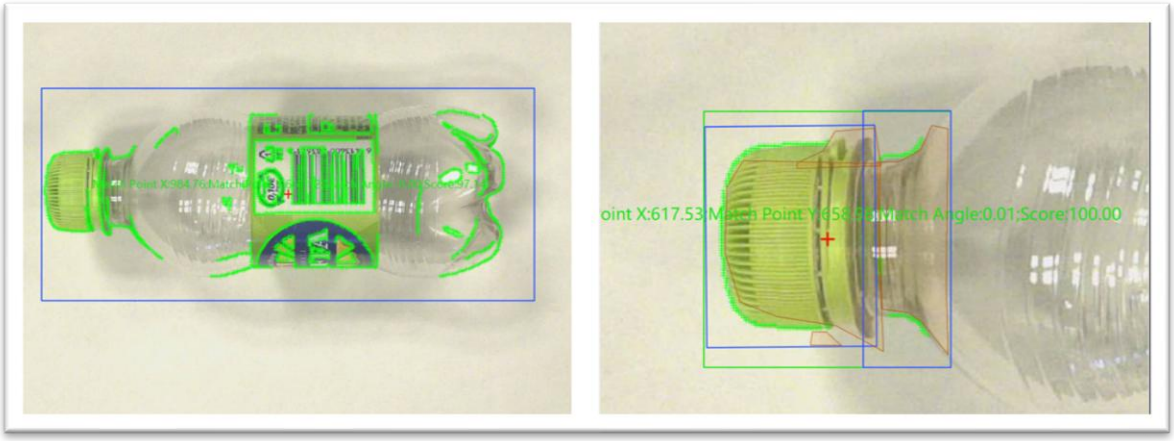

Figure 7. Experiment with translucent water bottle. On the left is the initial contour recognition, and on the right is the functional feature template. Screenshot from Hikrobot SCMVS -software, Veeti Suominen, 2026.

With translucent objects, the problem becomes how do you detect something that you can see through. No clear and continuous outline is formed as seen on the left side of Figure 7. Along with no clear outline, the algorithm finds contours in the background behind the translucent object, which of course are changing unexpectedly. In the case of a translucent object, detecting the entire object with traditional methods is quite impossible. The solution in such scenarios is often to stop looking at the whole object and instead teach the software to look for a specific non-translucent detail, like the bottle cap. As seen on the right side of Figure 7. the functional template uses the bottle cap and part of the bottle neck as a distinct and clear details for detection. Unwanted edges inside the bottle cap and neck were shielded out, leaving only the outline. This works well, even with multiple objects in the view, as shown in Figure 8., where the program detects two bottles successfully.

Figure 8. Experiment with translucent water bottles showing successful detection of two target objects through training the feature template for finding the bottle caps. Screenshot from Hikrobot SCMVS - software, Veeti Suominen, 2026.

4.2 Visual error related challenges

This section includes the experiments done regarding the challenges that visual problems can cause for traditional machine vision. First experiment is a white phone charger against a white background.

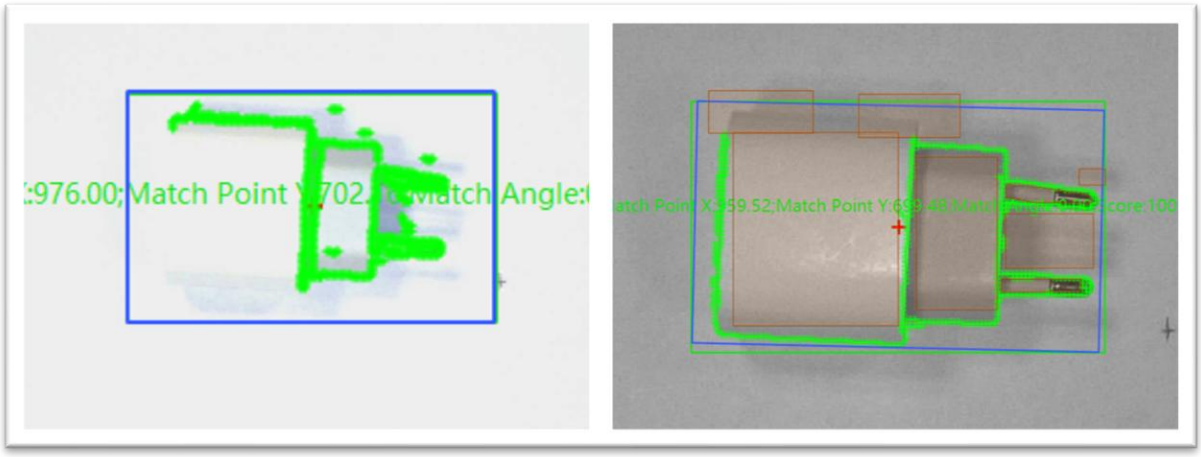

Figure 9. Experiment with light phone charger against a white background. On the left is the initial contour recognition, and on the right is the functional feature template. Screenshot from Hikrobot SCMVS - software, Veeti Suominen, 2026.

When a target object and its background share similar brightness, meaning that their grayscale values live in the same area on the histogram, the lack of distinct contrast creates a low-contrast problem. As seen in the initial contour recognition on the left side of Figure 9. traditional thresholding fails to find a stable and distinct edge or outline. The resulting contours are fragmented and noisy, not suitable for reliable detection. To overcome this, lighting conditions are manipulated to lower the exposure of the slightly reflective charger material, as well as adjusting the grayscale threshold and camera parameters. This results in an initially better contour recognition, with more robust and clear outlines. Through applying manual shielding areas, the program is forced to focus only on the stronger edges, resulting in a functional feature template shown on the right side of Figure 9. Of course, the easiest way to overcome a low-contrast problem is to change the background colour, but this might not always be possible. For example, in a situation where a charger production line can have chargers both black and white run under the same machine vision system, or in a situation where it is not seen beneficial to invest into a new different colour conveyor belt.

The next experiment is very similar, this time with a dark target object against a dark background.

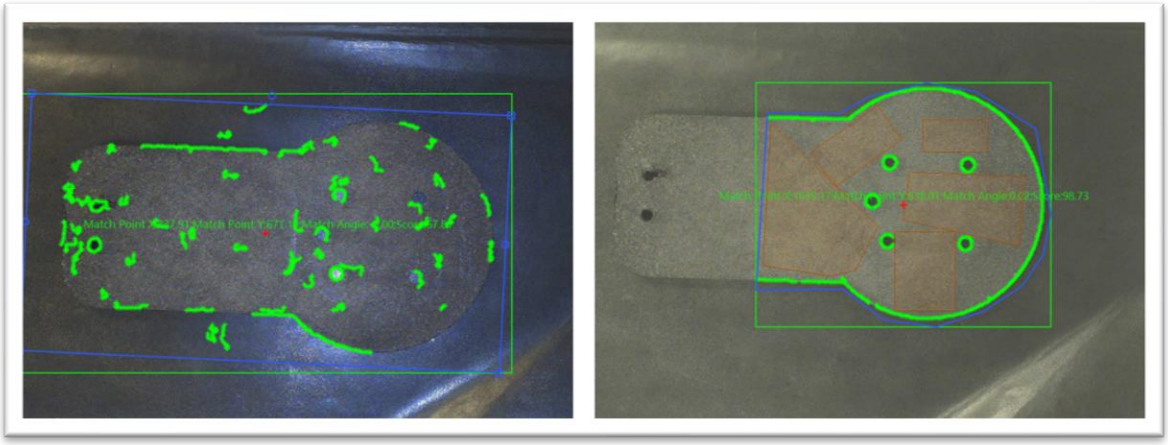

Figure 10. Experiment with dark plastic part against a dark background. On the left is the initial contour recognition, and on the right is the functional feature template. Screenshot from Hikrobot SCMVS - software, Veeti Suominen, 2026.

A similar low-contrast problem occurs when detecting dark objects on dark backgrounds. In these scenarios, the lack of a distinct outer edge is similarly missing, and internal surface textures, holes and scratches catch the light and show up as

contours more easily than on a white object. As seen in the initial contour recognition on the left side of the Figure 10., the algorithm is distracted by this internal noise and results in a fragmented contour that completely fails to track the true outline of the object. To overcome this, adjustments are made to the lighting conditions, and exposure and grayscale threshold parameters. Turning off the camera active lighting gets rid of the distinct highlights on the matte surfaces. Fine-tuning the grayscale threshold showed that forming a continuous outline from the faint gradient was not possible, so only a section of the object outline is used. Along with manual shielding of internal contours, except for the distinct holes, the process results in a stable and clean outline for a functional and reliable feature template, shown on the right side of the Figure 10. The main differences between this experiment compared to the light-on-light scenario, are how the dark surface catches the light and creates more distinct highlights, which are not visible if the object is also light or white. Also, the shadows are not as visible on the dark background and therefore do not create ghost edges as easily as on the white background.

4.3 Limitations and problems with 2D

A fundamental constraint of 2D machine vision is its inability to perceive depth, something that 3D systems can do. Because the camera flattens the 3D -world into a 2D -image, along with lens distortion, variations in the viewing perspective can severely distort an object's perceived shape and coordinates. The following experiments demonstrate how this constraint affects reliable detection and positioning.

The first experiment showcases how a 2D system is susceptible to geometric illusions caused by lens perspective and object height. To demonstrate this, wooden blocks were arranged on the table surface as shown in Figure 11.

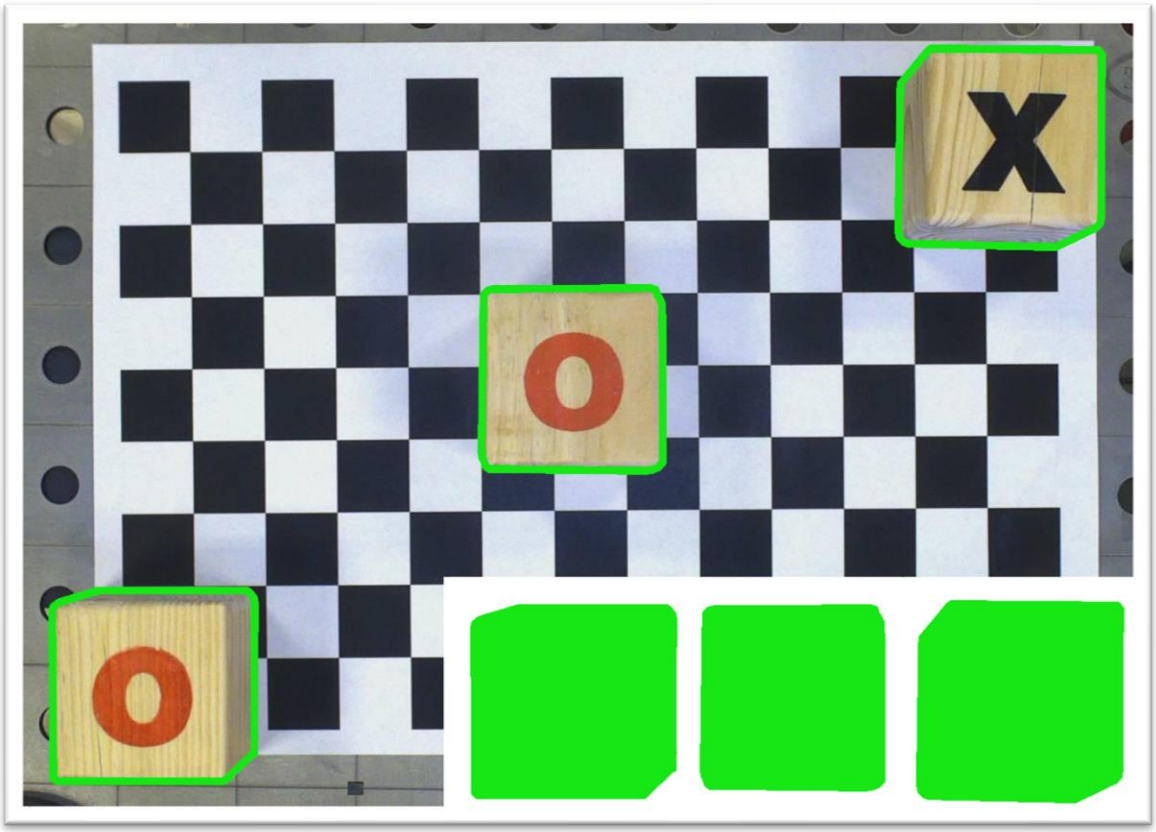

Figure 11. Experiment showcasing the distortion of perceived object shape due to perspective changes using wooden blocks. The edited green lines represent hypothetical outlines found through edge detection. Shape difference is highlighted with filled in outlines on the right-bottom corner of the image. Original image captured through Hikrobot SCMVS -software, edited in Photoshop, Veeti Suominen, 2026.

The green lines representing hypothetical contours are not from the software but edited in by hand. When the block is placed directly on the camera's optical axis, it only captures the top surface and is perceived as a square. However, when the same block is moved to the outer areas of the view, the cameras viewing angle changes. This perspective change reveals the sides of the block, and to the 2D software, the objects shape has fundamentally changed into a more complex polygon. This would cause the standard template matching to fail, if it is trained with the square outline of the centered block.

The next experiment showcases how variations in object height (Z-axis) artificially alter its perceived coordinates on the X-Y -plane. The feature template is trained to detect the "X" shape on the top of the wooden block, and not the whole block itself. The small cross in the middle of the "X" shape seen in Figure 12. indicates the center of the detected object's boundary box which the given coordinates are based on.

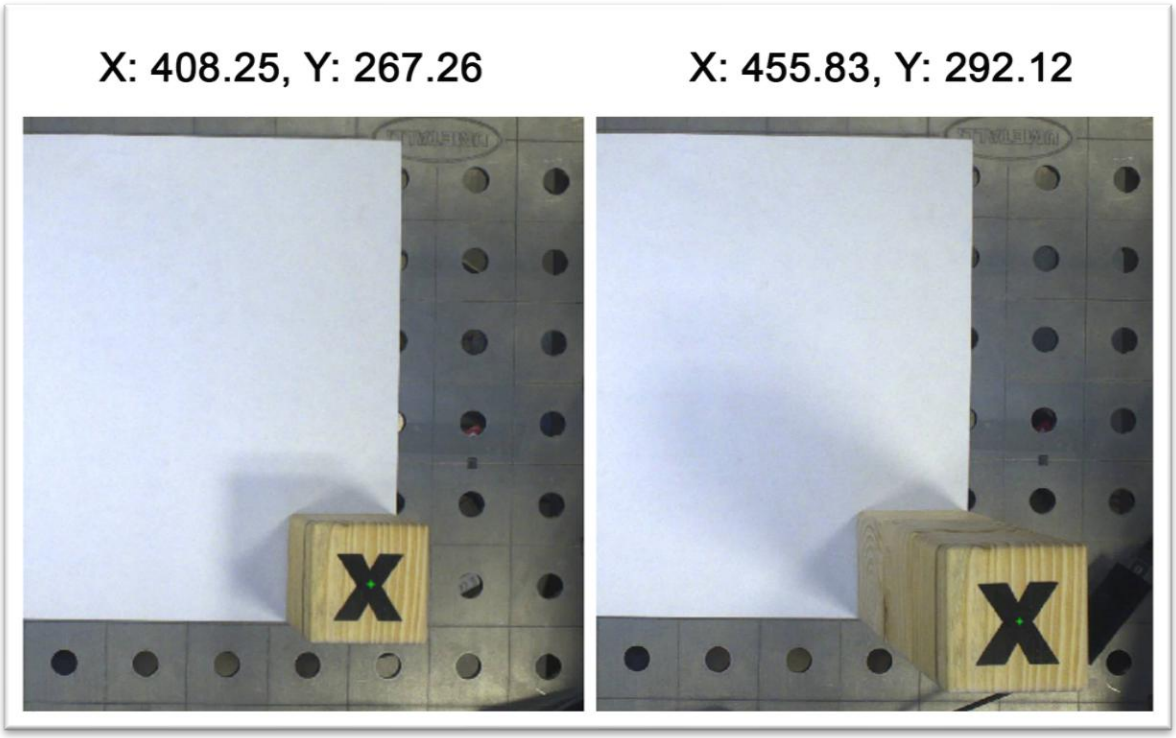

Figure 12. Experiment demonstrating how variation in z-axis height causes a parallax effect and artificially alters the X-Y plane coordinates. The coordinates extracted from the software are shown above each scenario. On the left side a single block is placed on the outer edge of the view, and on the right side 3 blocks are stacked to create height. Original screenshot from Hikrobot SCMVS -software, edited in Photoshop, Veeti Suominen, 2026.

Because the stack is positioned away from the camera's optical axis, raising the object closer to the lens induces a parallax effect. The perspective distortion causes the elevated X to be projected further outward across the 2D image. This can be seen by eye, but as well with the coordinates shown on Figure 12. where both coordinate values increase. In reality, the X-Y plane position of the X -shape is still the exact same as with the singular block. The greater the height difference, the greater the shift in coordinate values, supposing were on the outer edge of the camera view.

It is to be noted that this experiment demonstrates a situation where the object height changes, but the software is not recalibrated for the new X-Y plane. Meaning that reliable coordinates are possible at any height through calibration, but a continuously running application outputting reliable coordinates requires static height of the target object.

5 Discussion and summary of experimental results

5.1 Evaluating the boundaries of 2D machine vision

The primary objective of this experimental study was to evaluate the capabilities and physical limitations of a 2D smart camera using traditional detection algorithms. Rather than testing in ideal laboratory conditions, the experiments were designed to expose the system to real-world complexities that operators or developers might encounter, such as reflective surfaces, low-contrast environments, and geometric distortions. The results demonstrate how these physical and optical challenges manifest themselves within the software. This reinforces the statement made in the literature review that machine vision performance is fundamentally dictated by the quality of the input data, while software tools and methods can successfully compensate for poor data in some cases, the baseline reliability is dependent on the quality of the raw visual input.

5.2 Material and optical complexities in production

In industrial automation, material properties affect the success of part detection. Reflective or translucent materials cause the edge detection to interpret highlights and reflections as "ghost" edges or causes it to fail at finding continuous outlines entirely. Similarly, low-contrast environments produce fragmented and noisy contours due to faint edge gradients. As the fundamental machine vision principle "garbage in, garbage out" dictates, traditional rule-based algorithms cannot reliably process images where edges are obscured. While the experiments showed that these scenarios can be combated by adjusting grayscale thresholds and applying manual shielding, this introduces a significant lack of flexibility. In a production environment, such rigid parameter tuning and filtering means that even a minor change in ambient lighting or a slight shift in material quality or features could instantly break the detection template and -reliability.

5.3 The depth and perspective related vulnerabilities

A major weakness of 2D systems is that they cannot see depth. The tests showed clear visual errors caused by camera perspective and the system's inability to measure

height. Moving a block from the center to the edge of the camera's view revealed its sides, which completely changed its 2D shape in the software. Furthermore, making an off-center object taller caused a major perspective error. Even though the block did not move on the physical table, the distortion made the software report a large change in its X-Y -coordinates. In an automated robotic setup, this is a critical problem. If part heights change on a production line, the 2D camera will calculate the wrong locations, possibly causing a robotic arm to completely miss its target and risk damage to products or equipment. This problem can be attempted to be avoided through calculational and computational measures, but for a complex application with a variety of different parts, a 3D system should be considered.

6 Conclusion

This thesis examined the physical and algorithmic foundations of 2D machine vision systems for industrial part detection. Through a comprehensive literature review, the study first established the core principles of machine vision technology. The review traced the evolution of the field and detailed the essential components required for a functional system: lighting, optics, image acquisition, and image processing. A key theoretical focus was the difference between traditional rule-based algorithms—which rely on clear mathematical parameters like thresholding and edge detection—and modern deep learning methods. Ultimately, the literature highlighted that the success of a machine vision system is heavily dependent on the quality of the captured image, confirming the known "garbage in, garbage out" principle.

The practical experiments directly tested these theoretical limits by utlizing a rule-based 2D smart camera on real-world scenarios. Testing with reflective surfaces, translucent materials, and low-contrast environments proved that traditional algorithms struggle significantly when clear edge gradients are disrupted. While the software can be forced to work through strict parameter adjustments and manual shielding, this approach makes the setup highly rigid. In a real industrial environment, this lack of flexibility means that even minor changes in ambient lighting or material properties can cause the detection template to fail.

Furthermore, the experiments demonstrated the fundamental geometric limits of 2D cameras, specifically their inability to measure depth. The tests proved that moving an object to the edge of the camera's view, or changing its physical height, causes severe perspective distortion. This distortion alters the object's perceived 2D shape and results in incorrect coordinate calculations. In automated robotic tasks, these errors can cause a robot to physically miss its target, proving that reliable 2D systems require strict control over object placement and height.

In conclusion, traditional 2D machine vision remains a fast, efficient, and cost-effective tool for industrial tasks where the environment, lighting, and object presentation can be strictly controlled. However, its heavy reliance on good conditions limits its reliability in more complex situations. As established in both the literature review and the practical

tests, overcoming these physical constraints can be seen in modern industrial automation as it is increasingly utilizing deep learning models for better flexibility, adaptability and complex applications, as well as deploying 3D vision systems to avoid the inherent depth and perspective constraints.

References

Golnabi, H., Asadpour, A., 2007. Design and application of industrial machine vision systems. Robotics and Computer-Integrated Manufacturing 23, 630–637. https://doi.org/10.1016/j.rcim.2007.02.005

Hornberg, A., 2017. Handbook of Machine and Computer Vision: The Guide for Developers and Users, 1st ed. Wiley. https://doi.org/10.1002/9783527413409

Kim, S., Nguyen, T.P., Yoon, J., 2026. A Systematic Review of Machine Vision Applications in Factory and Manufacturing Processes: From Quality Control to Predictive Diagnostics. Int. J. of Precis. Eng. and Manuf.-Green Tech. 13, 745–775. https://doi.org/10.1007/s40684-025-00769-2

Smith, M.L., Smith, L.N., Hansen, M.F., 2021. The quiet revolution in machine vision - a state-of-the-art survey paper, including historical review, perspectives, and future directions. Computers in Industry 130, 103472. https://doi.org/10.1016/j.compind.2021.103472

Sun, Y., Sun, Z., Chen, W., 2024. The evolution of object detection methods. Engineering Applications of Artificial Intelligence 133, 108458. https://doi.org/10.1016/j.engappai.2024.108458

Szeliski, R., 2011. Computer Vision: Algorithms and Applications, Texts in Computer Science. Springer London, London. https://doi.org/10.1007/978-1-84882-935-0

Tzampazaki, M., Zografos, C., Vrochidou, E., Papakostas, G.A., 2024. Machine Vision—Moving from Industry 4.0 to Industry 5.0. Applied Sciences 14, 1471. https://doi.org/10.3390/app14041471

Wang, C., 2022. The Application of Machine Vision in Intelligent Manufacturing. HSET 9, 47–50. https://doi.org/10.54097/hset.v9i.1714

Yang, D., Peng, B., Al-Huda, Z., Malik, A., Zhai, D., 2022. An overview of edge and object contour detection. Neurocomputing 488, 470–493. https://doi.org/10.1016/j.neucom.2022.02.079

Yuheng, S., Hao, Y., 2017. Image Segmentation Algorithms Overview. https://doi.org/10.48550/ARXIV.1707.02051

Zhang, C., Xu, X., Fan, C., Wang, G., 2021. Literature Review of Machine Vision in Application Field. E3S Web Conf. 236, 04027. https://doi.org/10.1051/e3sconf/202123604027