Bachelor Thesis

TGMEKh20, 15hp

Reusage classification of damaged Paper Cores using Supervised Machine Learning

Halmstad 2023-05-04

Victor Larsson, Max Elofsson

HÖGSKOLAN
I HALMSTAD

Abstract

This paper consists of a project exploring the possibility to assess paper core reusability by measuring chuck damages utilizing a 3D sensor and using Machine Learning to classify reusage. The paper cores are part of a rolling/unrolling system at a paper mill whereas a chuck is used to slow and eventually stop the revolving paper core, which creates damages that at a certain point is too grave for reuse. The 3D sensor used is a TriSpector1008 from SICK, based on active triangulation through laser line projection and optic sensing. A number of paper cores with damages varying in severity labeled approved or unapproved for further use was provided. Supervised Learning in the form of K-NN, Support Vector Machine, Decision Trees and Random Forest was used to binary classify the dataset based on readings from the sensor. Features were extracted from these readings based on the spatial and frequency domain of each reading in an experimental way. Classification of reusage was previously done through thresholding on internal features in the sensor software. The goal of the project is to unify the decision making protocol/system with economical, environmental and sustainable waste management benefits. K-NN was found to be best suited in our case. Features for standard deviation of calculated depth obtained from the readings, performed best and lead to a zero false positive rate and recall score of 99.14%, outperforming the compared threshold system.

Sammanfattning

Den här rapporten undersöker möjligheten att bedöma papperskärnors återanvändbarhet genom att mäta chuckskador med hjälp av en 3D-sensor för att genom maskininlärning klassificera återanvändning. Papperskärnorna används i ett rullnings-/avrullningssystem i ett pappersbruk där en chuck används för att bromsa och till sist stoppa den roterande papperskärnan, vilket skapar skador som vid en viss punkt är för allvarliga för återanvändning. 3D-sensorn som används är en TriSpector1008 från SICK, baserad på aktiv triangulering genom laserlinjeprojektion och optisk avläsning. Projektet försågs med ett antal papperskärnor med varierande skador, märkta godkända eller ej godkända för vidare användning av leverantören. Supervised Learning i form av K-NN, Support Vector Machine, Decision Trees och Random Forest användes för att binärt klassificera datasetet baserat på avläsningar från sensorn. Features extraherades från dessa avläsningar baserat på spatial och frekvensdomänen för varje avläsning på ett experimentellt sätt. Klassificering av återanvändning gjordes tidigare genom tröskelvärden på interna features i sensorns mjukvara. Målet med projektet är att skapa ett enhetligt beslutsprotokoll/system med ekonomiska, miljömässiga och hållbara avfallshanteringsfördelar. K-NN visades vara bäst lämpad för projektet. Featuerna representerande standardavvikelse för beräknat djup som erhållits från avläsningarna visades vara bäst och leder till en false positive rate lika med noll och recall score på 99.14%, vilket överpresterade det jämförda tröskelsystemet.

Key-words

Machine Learning, Supervised Learning, classification, 3D sensor, TriSpector1008, SICK, paper core, paper mill, KNN, Support Vector Machine, Decision Trees, Random Forest

Acknowledgments

We would like to express our sincere gratitude to everyone involved in helping us with this Bachelor Thesis.

First and foremost, we would like to thank our supervisor, Björn Åstrand for his invaluable support and guidance during the project. His expertise has helped us shape this project and make the research more focused.

We would also like to thank Jens Lundström for his expertise and help with the understanding of Machine Learning and AI.

Secondly, we would like to thank Nils Strandh and CoreLink AB for providing us with the project idea, the paper cores and everything needed for the collecting of data. But also for guiding us through the world of paper manufacturing.

Last but not least we would like to thank Rikard Andersson and SICK for providing us with the TriSpector1008 and the support needed to understand it.

Table of content

1. Introduction

In the last 300 years, humankind has made significant advancements, especially in industrial manufacturing. Industrial manufacturing is among the sectors where artificial intelligence and machine learning enable an abundance of opportunities [1]. IIoT (Industrial Internet of Things), providing wireless connectivity for real-time manufacturing data collection and processing, has resulted in the culmination of the fourth industrial revolution, Industry 4.0 [1]. Industry 4.0 aims to provide real-time intelligent, interoperable and autonomous industrial manufacturing environments. Scientific literature revolving around Industry 4.0 were explored using unsupervised machine learning and text mining. Which identified 31 research and application issues, categorized into a five-level hierarchy: "1) infrastructure development for connection, 2) artificial intelligence development for data-driven decision making, 3) system and process optimization, 4) industrial innovation, and 5) social advance" [2]. This project covers categories 2 and 3, data-driven decision-making and system/process optimization.

1.1 Purpose

The goal is to explore the possibility of automatically assessing paper core reusability in a rolling/unrolling system at a paper mill with the help of sensors and machine learning. A decision between approved reuse or not is to be self-learned using a Machine Learning algorithm. The purpose of the project is to better distinguish between usable and unusable paper cores. Paper cores evaluated unfit for further use are today completely discarded. The paper core in question costs 80 SEK per meter, in total 250 SEK per core. Each day 15-30 cores are discarded, roughly 2 every hour at a total cost of 7000 - 10 000 SEK per day. Machine Learning could extend the core lifetime, give a clear indication of reuse, decrease the risks from core complications, such as human injury and machine damage. But primarily be more cost efficient and have more sustainable resource consumption. The goal and the purpose is to explore a base for construction of a new unified decision-making protocol/system grounded in economic, environmental, and sustainable waste management aspects.

1.2 Sensing system for paper core damages

Today, a paper core is reviewed for damage using a laser triangulation sensor. The core is reviewed and based on a set of threshold values, approved for further use, or sorted out to be discarded. The process is seen in *Figure 1*. The threshold values are based on the amount of damage and deformation the paper core can have without it being a risk when rolling and unrolling paper on the core.

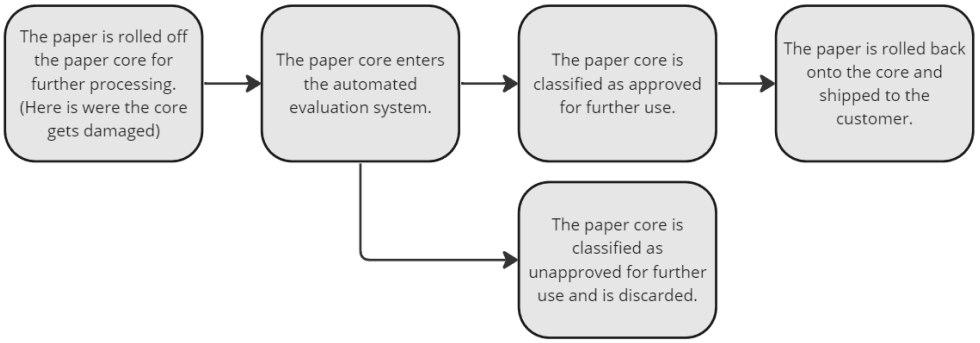


Figure 1. A simplification of the classification process used for reviewing the paper cores.

Chuck damages on cores

In many paper mills the paper is rolled onto the paper core to cool down. After the paper cools, it gets rolled off the core for further processing. A chuck connected to an electric motor is used to roll the paper up and off. These chucks create marks in the paper cores, as seen in *Figure 2*. To keep costs down, the paper cores used for rolling and unrolling the paper are reused and discarded first when the damage becomes too substantial and might result in the core coming loose from the chuck during handling.

Figure 2. One possibility of core damage from the chuck on the paper core

Paper cores

Paper cores are long cardboard tubes, in our case 3 meters by 12 inches, that are created in such a way that they can take tremendous pressure from the outside. This is done by gluing or laminating layers of paper together using an adhesive. The layers of paper can consist of recycled paper, paper-adhesive composites, cardboard or kraft paper. [3]

These cores are most commonly used for rolling up the paper for transport and cooling in paper industry factories but also for rolling up tapes, jute and fabrics [3].

System and sensor

The current system uses a 3D image sensor based on laser triangulation to evaluate from threshold values if a paper core is reusable or not from threshold values. The sensor is a TriSpector from *SICK*, model TriSpector1008 [4]. The TriSpector consists of an integrated laser line projection and an optical sensor (camera), which captures a series of profiles as the object is moved through the projection and builds a shape profile of the object [5]. *SICK*'s software interface *SOPAS* is used to set up the unit, threshold values, image capturing rate and capturing field [6].

1.3 Problem Statement

To reach the goals of the project, three major problem statements were created, dividing the goal into measurable and answerable sub-goals. Each statement handles a specific part of the project. Statement one covers the intrinsic features of a reusable or unusable paper core. Suggesting if a clear distinction can be found and what the distinction is between both. Statement two covers the sensing and capturing technique of the used and available sensors. Evaluate the precision and repeatability of the used sensor to establish an understanding and knowledge of whether readings of the paper cores and damages are sufficiently detailed. Statement three questions what algorithms and whether Machine Learning classifies reuse better than thresholding.

1. What specifies a paper core that is approved for further use?

2. Do the sensors in the current system yield a sufficiently detailed description of features/parameters for continued use of a paper core?

3. What or which machine learning algorithm and models are best suited for our case of decision-making in order to classify more accurately than a threshold value system?

2. Background

The current system uses a TriSpector1000, which is a 3D imaging vision sensor based on laser triangulation. Approval of re-usage is evaluated and decided through a set of threshold values set up in the sensor software. *Machine Learning* (ML) could be used instead of threshold values to approve re-usage.

2.1 Vision Sensors

Vision sensors use captured images from a camera to determine presence, orientation and accuracy of parts and use a combination of image acquisition and image processing [7]. Vision sensors occur in two models, monochrome or color. Captured images are converted to an electrical signal by the light-receiving element, which determines the intensity information belonging to each pixel [8]. The monochrome model identifies intensity ranges between white and black pixels, unlike the color model which separates light information into three colors (RGB). Vision sensors/systems are further divided into three categories 1D, 2D and 3D. 1D vision analyzes a digital signal one line at a time instead of the whole picture at once [9], assessing variance between groupings of recent lines compared to earlier groupings [10]. Commonly used in detecting and classifying defects on manufactured materials in a continuous process, paper, metals or plastics [10]. 2D vision typically consists of comparing contrasts in the captured two-dimensional map of intensity. Used for tasks such as barcode reading, character recognition, label verification and presence detection. 2D has parallax, depth of focus, ambient light and contrast variations as limitations [9]. 3D usually comprises multiple cameras or laser displacement sensors mounted in different locations in combination with triangulation methods. Providing X, Y and Z data of a target regardless of environmental conditions. 3D vision is used broadly, measuring thickness, height and volume, detecting surface or assembly defects, object scanning and digitization [9].

2.1.1 Distance measurement sensors, 3D vision sensing

3D sensing is a depth-sensing technology capturing the objects real-world length, width and height [11]. Sensors scan both the surface and depth of an object [8], meaning that the sensors can analyze contents inside of a packaging or, in the case of this project, paper core surface damage. Shape acquisition through reflection sensors can be subdivided into optical and non-optical sensing. Optical sensing captures measurement information carried by the reflected light. Non-optical include acoustic sensors (ultrasonic, seismic), electromagnetic (infrared, ultraviolet, microwave

radar, to name a few) and others. These typically measure distances by measuring the time required for the energy to bóunce back from the object [12]. Optical methods operate in either active or passive form, projecting or acquiring electromagnetic energy onto/from an object, then recording the transmitted or reflected energy.

To obtain 3D imagery any given active or passive method belongs to one of three main techniques: geometrical, time of flight (TOF) or interferometry [13]. The first method exploits the geometrical relationship between the target and detection system. Examples of such techniques are passive and active triangulation, structured light and depth from defocus. These techniques can measure distances from about a millimeter to many kilometers [13].

A TOF method measures distance by measuring a light signal's travel time between the target and light detector. In a direct TOF setup time is measured by a clock, while in an indirect TOF setup time is inferred from, for example, the phase relationship between intensity-modulated emitted and detected light [13]. This technique is similar to interferometry, though range is not constrained by the wavelength of light but instead by the wavelength of modulation, which is adjustable to specific needs. An example of a TOF system is Light Detection And Ranging (LIDAR), where both direct and indirect versions exist. Timer jitter of the clock and the photodetectors used to control it limit the smallest measurable distance, while the strength of the returned light limits the largest indirect TOF. In indirect using pulsed illumination, factors such as illumination pulse duration limit the minimum and maximum ranges.

Interferometry is a technique using the phenomenon of interference of waves [14], exploiting the nature of light. The wavelength of visible light is in the 400 -700 nm range. Devices based on this principle and using visible light, can probe depth on a comparable scale [13]. By splitting one light beam into two, an interference pattern can be formed when these two beams superpose. Small differences in the optical paths (distance traveled) between the two beams can be detected (as these differences will produce noticeable changes in the interference pattern).

The most common technique used in commercial 3D sensors is Laser Triangulation [15], also used in this project.

Laser Triangulation

Laser triangulation is an optical triangulation-based 3D technique involving the projection of a ray of light onto an object and detecting the reflections by an optical sensor. *Figure 3* shows a system configuration. Points O_P and O_C are the exit and entrance pupil of the laser source and camera, with the

mutual distance d. The points' optical axes Z_P and Z_C form the angle of projection α [12]. Knowing this, the point's position in the space can be determined [15]. In the case of a single-point source, the object must be scanned vertically and horizontally to obtain the depth. Laser stripe (line) projection is faster and can obtain the depth for all points of the line simultaneously. [15]

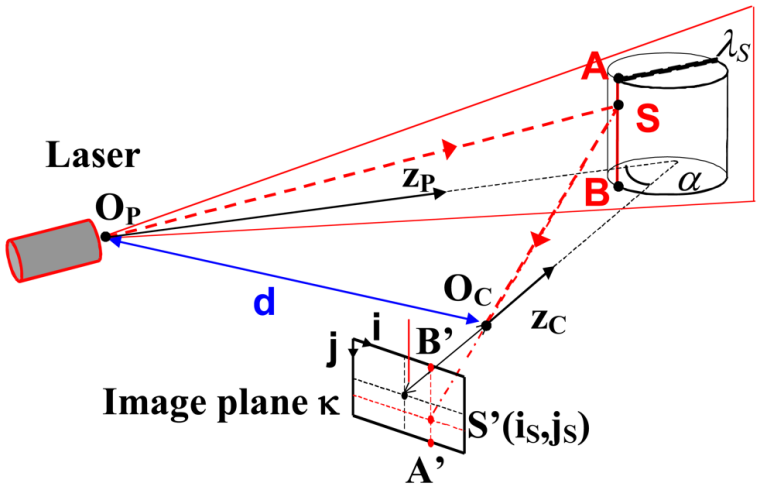


Figure 3 - Schematic of the triangulation principle, adapted from [12], reused in accordance with MDPIs open access policy.

One significant advantage of laser triangulation is the accuracy and its relative insensitivity to illumination conditions and surface texture effect [12]. The method is limited in cases of very high precision and accuracy and is plagued by speckle and scanning inability of reflective surfaces [15].

2.2 Today's sensor setup, TriSpector1000

The sensor used has previously been used to measure the dynamic response of a laser forming process, combining measurements to approximate a surface [16]. Automate height assessment in tree nurseries, store and acquire real-time 3D point-cloud data of seedlings [17]. Record the topography of a bed containing plants and sand designed to test dune development and stabilization [18]. In an automated quality inspection as part of an analysis of the potentials of collaborative robotics in industry, part of industry 4.0 [19]. The applications of the sensor are mostly in surface measuring, recreating a 3D representation of the scanned environment. The seedling measurement in [17] using the TriSpector1060 generated highly repeatable height measurements below 5*mm*.

2.3 Machine Learning

The project will use Machine Learning to sort the approved paper cores from the unapproved ones. Machine learning is a method where a computer system is trained to make decisions and take actions without explicitly being programmed for these decisions or actions [20]. The use of machine learning is widespread in today's society and is used, for example, in areas like Healthcare and economics [21], but the most well-known use is in search engines like Google [20]. Machine learning can be divided mainly into three methods. Supervised Learning, Unsupervised Learning, Reinforcement Learning [22].

The benefit of a machine learning system towards the project is that the operator will not need to be trained in what an approved and unapproved paper roll is. Instead, the system will only need a labeled training dataset to train on to decide if a paper core is approved for further use or should be discarded.

Supervised Learning

When using supervised learning, the system is trained using an algorithm and a training dataset. Once the system has been trained how to act, using the training dataset, the system will do these actions automatically [20]. Within this learning method, the most used techniques are *Regression*, where the machine learning system estimates the relationships between variables and is often used for predictions, and *Classification*, where the machine learning system draws conclusions from the training dataset and uses this to classify the new input data [22].

Unsupervised Learning

With unsupervised learning, there is no guidance from either labels or human interaction. Instead, the system finds correlations between the data points and then groups them or arranges them in a certain way according to these correlations. Within this learning method, the most used technique is Clustering, where the data points are grouped based on the correlations in the data [22].

Reinforcement Learning

In reinforced learning the output is not a state or a value but rather a series of actions leading towards an objective or goal. Thus a good decision or action is not defined by the action itself but rather if the action is part of a good policy that leads to the end goal. These policies are generated by having the system learn from previous good action sequences. Reinforcement Learning is often used in games since each turn has many possible moves. Where a single bad move could still lead to a victory if it is

part of a good strategy and a single good move could still lead to defeat if it is part of a bad strategy [22], [23].

Deep Learning

Deep learning is a subdomain within Machine Learning that uses Neural Networks to self-learn patterns in a dataset to make predictions, like classification, based on these patterns. In simple machine learning, the features used for finding the patterns in the dataset are extracted manually and then classified using a machine learning algorithm. Where in Deep Learning, these features are self-learned in the network. This enables the Deep Neural Networks, in many cases, to have a higher accuracy than a regular machine learning algorithm. However, deep learning requires great computing power and big datasets to perform well [22].

2.4 Machine Learning Classification in an industrial environment

With Industry 4.0 Machine Learning has deeply impacted the manufacturing industry. Its predictive insights offer a pathway for a decision support system for various industrial tasks including process optimization and predictive maintenance but also intelligent and continuous inspections [24]. Similar to this project, the scientific paper by Chen *et al*.[25] shows that machine learning can be used for the classification of welding defects. The supervised learning technique used was shown to be highly robust, precise and reliable for ultrasonic defect classification. This shows that classification is possible and usable within industrial processes and as a part of Industry 4.0.

3. Method

The project consists of three major problem statements seen in *Chapter 1.3*. These statements run parallel to some extent but have parts where milestones must be achieved before the rest of the project can proceed. To work effectively and keep within the timeframe these problem statements are divided into four parts. These parts are in turn divided into one methodology and one implementation part. A literature study is used to gather scientific literature and base each part upon. Following are the headers and summaries of the contents of the four parts.

Evaluate the current system - to ensure the current system is able to provide enough input data for the algorithms to be able to classify.

Collecting the dataset - since there is no previous data collected, the dataset that is to be used for training and validation of the model needs to be collected.

Image Processing and Feature Extraction - to extract features from the depth images needed for *Machine Learning* models.

Evaluating Algorithms - to ensure the best algorithm is used for the specific task, four classification algorithms will be tested and evaluated.

3.2 Evaluate the current system

3.2.1 Sensors

To evaluate the current system, studies surrounding vision systems and sensors will be read and collected through parts of a literature study. Methods found concerning 3D imaging are compiled and evaluated in the aspects of its areas of application reflected on the case in this project. One method is then selected to be chosen in the system. Sensors using this method will be compared through certain qualifications consisting of robustness, established brand and size.

A repeatability test will be carried out on the current sensor, Trispector 1008, where the same core will be measured several times from the same starting position. In order to verify the repeatability of the sensor measurements.

In order to verify that the sensor's accuracy is capable of properly measuring the damages of each core. The damages will be measured using the images gathered from the sensor and then calculated by transforming the pixels into millimeters. The size of the damages will then be compared to the sensor's accuracy.

3.2.2 Baseline system

In order to compare the machine learning system a simple threshold value system will be constructed similar to the one used by SICKs SOPAS.

The system will have two thresholds. One for the maximum height and one for the maximum depth the paper core is allowed to differ from the surface of the core. If a measurement is outside either of these threshold values the core is classified as unapproved.

The threshold values will be calculated by finding an approved paper core's maximum height and depth value and an unapproved paper core's minimum height and depth value. These will create two spans of values, one for height and one for depth, where the threshold values can be chosen and tested. This will be done manually by trial and error using a train test split of the dataset. Both to make it more comparable with the Machine Learning models and to validate that the threshold system is not overfitted.

3.3 Collecting dataset

To collect the data regarding what is perceived as an accepted paper core, a test rig will be used. The test rig consists of the *TriSpector1000* model 1008, a modified paper core cutter and an *Incremental encoder DBS36/50* (*DBS36E-BBCP02048* [26]) from *SICK* with a 30mm rubber wheel attached to it. The paper core cutter has been modified by canceling the saw blade action. This will ensure that the paper core is not destroyed when the cutter is used to rotate the paper core. The cutter is also modified by fastening holds for the TriSpector and the *Incremental encoder DBS36/50.* The *TriSpector* is positioned in a way that it reads the inside surface of the core at one of the core endings and the rubber wheel connected to the encoder is positioned in a way that it connects with the outside surface of the other end of the same core. The encoder is connected to the *TriSpector* to register the rotational speed of the core during reading.

The paper core is loaded into the test rig, rotated one revolution by hand, and read by the *TriSpector*. Each reading is then saved as a PNG-file and labeled with one of two labels, approved or unapproved, in order to mimic the current system where a paper core never enters the system unused. Thus the system only needs to classify it as either approved for further use or unapproved for further use. This will create a dataset suitable for *Binary Classification*. Which in our case could result in the dataset being unbalanced [27], meaning more samples of approved than unapproved.

An approved core is a core that is approved for further use. Either the damages are minor and will not disrupt the production line. An unapproved core is a core that can possibly disrupt the production line if continued use is allowed. The labels of the cores measured have been decided by the supplier

of the provided cores, which was done by threshold values and an ocular review from trained personnel.

3.4 Image Processing

Each reading (PNG-file) contains two parts, one third light intensity and two thirds depth. The images are in the format of 7491x192 square pixels, 0.38x0.38 [mm], set by the settings in SOPAS (seen in *Appendix [3]),* capturing one rotation 957,5mm and 74mm wide. Light intensity and depth are separated by splitting the images. Light intensity is the first 2497 rows and depth the last 4994 rows. The 16-bit depth parts are then decoded by adding pairs of pixels in the rows, halving the height but keeping the same width. The new pixel values consist of the sum of the leftmost pixel and the rightmost pixel multiplied by 256 for each pair, as seen in *Figure 4* below.

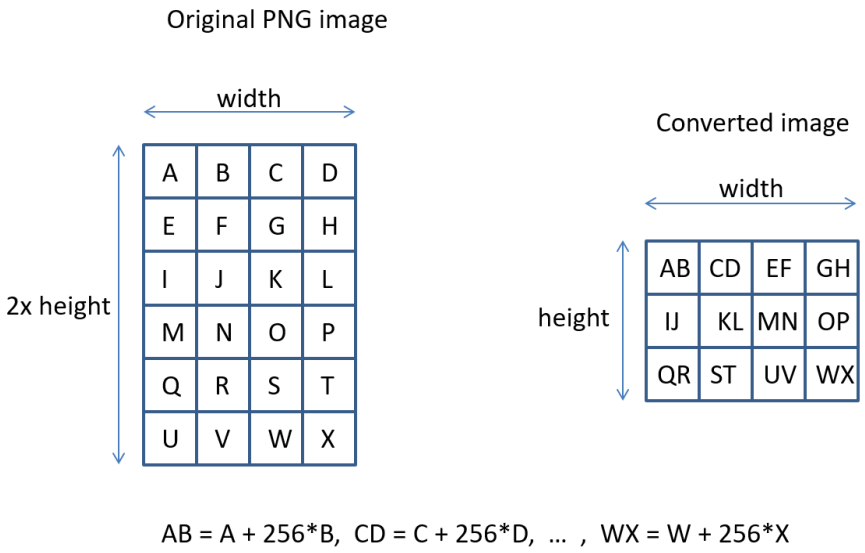


Figure 4. Depth image decoding protocol

Within the metadata of each reading (PNG file), stored as tEXt tags are values corresponding to the sensor's origin X0, Y0 and Z0 and scaling in the directions Dx, Dy, Dz. These scalar values enable conversion into real-world values, seen in *Appendix [18]*.

$$X_{world} = X0 + X_{pixel\ index} * Dx \quad (3.1)$$

$$Y_{world} = Y0 + Y_{pixel\ index} * Dy \quad (3.2)$$

$$Z_{world} = Z0 + Z_{pixel\ value} * Dz \quad (3.3)$$

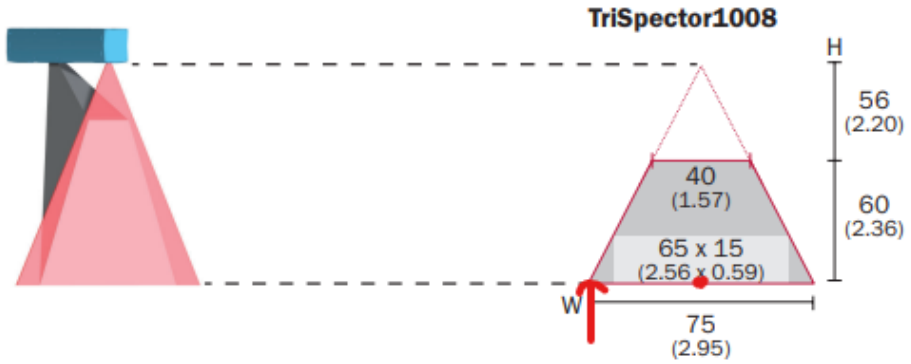


Figure 5. Typical field of view, TriSpector1008. Origin origo = dot, moved origo = arrow

X0, Y0 and Z0 are used to move origo from its origin marked by the dot in *Figure 5* to the bottom left corner of the field of view. Dx, Dy and Dz correspond to changes in real-world values in accordance with pixel indexes and scalar values shown in Equ. (*3.1, 3.2, 3.3)*. The origin of origo is 116 mm from the camera to the furthest point of field of view, a pixel value converted to real-world depth (Z-axis) equaling 13 mm means that point of the object is 103 mm from the camera.

3.5 Feature Extraction

Experiments on both the spatial and frequency domain related to the depth parts will be done to extract features. The spatial domain, which is two-dimensional matrices of grayscale pixel intensities of each image/depth part, is used in the first experiments since SICKs software SOPAS uses the real-world values corresponding to the spatial information in each reading (PNG file) to evaluate re-usage. A theory arose surrounding the Machine Learning algorithms, based on real-world values explained above, having difficulties differentiating between approved and unapproved since both classes of paper cores have very similar damages. For that reason experiments on the frequency domain, roughly meaning rate of change in pixel values, were done in belief of better features and performance since the damages reoccur with certain frequencies and also therefore certain recurring changes in pixel values.

Spatial domain

To extract features from the spatial domain meaning real-world depth measurements (z-axis values), statistical measures will be used to compress data from all pixels throughout the columns of the images. Columns will be used since the damages are present on the x-axis and get repositioned on the y-axis in new readings of different paper cores.

Frequency domain

To extract features from the frequency domain meaning the spectrum of frequency components of the pixel values in columns of the depth parts.

Fast Fourier Transforms are to be used, compressed by statistical measures to create a reasonable feature space in accordance with the dataset.

3.6 Evaluating Algorithms

The No Free Lunch theorem states that there is no single machine learning algorithm that is the most accurate in any domain [23]. Because of this, four different Machine Learning Algorithms will be tested. Since the dataset used for training, testing and validation already will be labeled and since the system only needs to make a binary decision whether the paper core is approved or unapproved for further use, Supervised learning and, more specifically, classification are suitable techniques for the project.

These Algorithms are:

k-NN

k-NN is a Supervised learning algorithm and classifies each data point in regard to the label of the k-Nearest Neighbors. Where *k* is a discrete variable with a value greater than zero. k-NN is considered a simple and easy-to-understand algorithm and is recommended for image recognition[28].

When using k-NN, the algorithm is first trained with labeled data. When a test sample is classified the Euclidean distance between the test sample and the training samples is calculated using *Equ. (3.4)*, where x_i is the input sample of n total number of samples ($i = 1, 2, 3, \cdots, n$) with p total amount of features ($x_{i1}, x_{i2}, x_{i3}, \cdots, x_{ip}$). x_i is compared to sample x_l ($l = 1, 2, 3, \cdots, n$). The input sample will then be classified with the most frequent label of the *k* closest training samples [29].

$$d(x_i, x_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \cdots + (x_{ip} - x_{lp})^2} \qquad (3.4)$$

Linear Support Vector Machine

When using a Support Vector Machine a hyperplane is defined as a decision boundary using a training dataset. A hyperplane is a plane or line that is $n - 1$ dimensions, where n is the number of features. If there are only two features the hyperplane is just a line. In an n-dimensional space, there are many ways a hyperplane could be used as a decision boundary. The algorithm will therefore try to find a hyperplane by iteratively generating hyperplanes that best separate the classes. The hyperplane that most correctly separates the classes will then be chosen.

There are two types of Support Vector Machines, Linear Support Vector Machine and Non-linear Support Vector Machine. Linear Support Vector Machines are used for classifying data that can be split into two separate

classes with a straight line. Otherwise, the Non-linear Support Vector Machine is used.

The algorithm's accuracy and performance are independent of the dataset size but rather the number of training cycles [30]. This makes it a good candidate for the dataset used in the project.

Decision Tree Classifier

The Decision Tree Algorithm can solve either a classification or a regression problem by continuously splitting the data based on a certain parameter. When used for Classification, the decision parameter is categorical, meaning it is either yes or no [28].

A tree is composed of decision nodes, leaves and branches connecting all the nodes and leaves together. Every decision node has a test function with a discrete output that labels the branches. When given an input, starting from the root node, the test function for the current node is applied and one of the branches is taken based on the outcome. This is repeated until a leaf node is reached and the class of the leaf is applied to the input data. When training a Decision Tree it never backtracks and will generate new decision nodes further on, even when there might be alternatives, creating unnecessary subtrees. The process of removing these unnecessary subtrees is called pruning[23].

Decision Trees are easy to interpret and have a good generalization ability [28], [30]. Which makes the algorithm suitable for binary classification.

Random Forest

If not pruned correctly a Decision Trees might overfit. Which is why Random Forest was developed. Random Forest is a method that trains a number of Decision Trees and returns the class with the majority of all the trees in the ensemble. Since Random Forests are fast, easy to interpret, do not over-fit, and have no parameters to manage, they are the winner of many classification problems [30]. This makes it a given candidate for the project.

Scoring and Evaluating the Machine Learning Models

When measuring the performance of a binary classifier, common measurements are *precision*, *specificity* and *recall,* also called *sensitivity* [23]. Precision measures the amount of true positives of all the predicted positives while Recall measures the amount of true positives predicted of all the labeled positives. Specificity measures the amount of true negatives of all labeled negatives. In a paper production plant, a paper core coming loose from the chuck could lead to a stop in production and, in the worst cases, even human injuries. That is why the classification system must have a *false positive rate* equal to zero to ensure that no paper core is unapproved and

classified by the system as approved. However, it is still essential that the *recall* value is kept reasonably high to keep the system from sorting out paper cores that could be reused without the risk of coming loose from the chuck, thus increasing the cost and environmental impact. This is visualized by a Confusion Matrix in *Figure 6*.

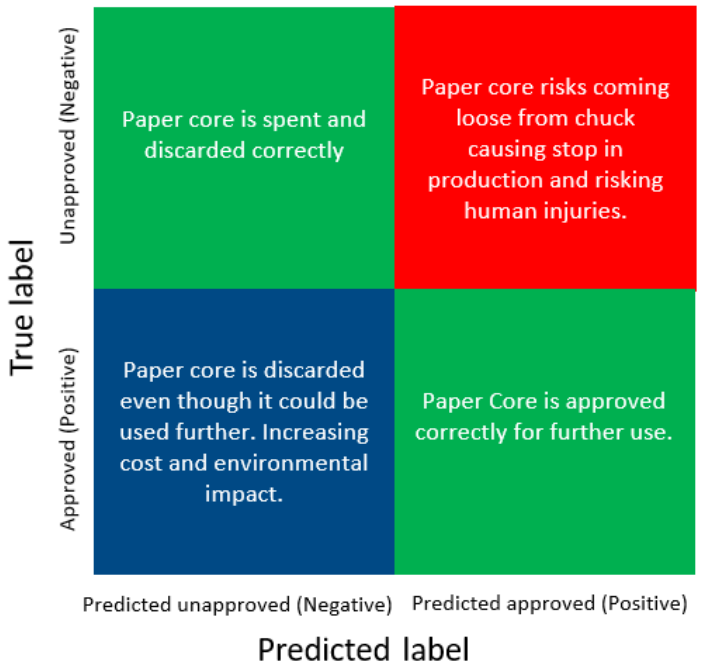


Figure 6. Confusion Matrix explaining a paper cores different classifications

The *false positive* rate is calculated using the *false positives* and the *true positives* [23].

$$\frac{False\ Positive}{False\ Positive + True\ Positive} = False\ Positive\ Rate \quad (3.5)$$

The Recall is calculated using the True Positive and False Negatives [23].

$$\frac{True\ Positive}{False\ Negative + True\ Positive} = Recall \quad (3.6)$$

The Score is calculated by making sure the False Positive rate is equal to zero, giving it the boolean expression of True which is equal to one, and then multiplying it with the Recall value.

$$Score = (FalsePositiveRate \equiv 0) \times Recall \quad (3.7)$$

3.7 Implementation

3.7.1 Evaluation of the current system

Sensor

To verify the sensor's repeatability, one approved roll was measured ten times from the same starting position. The mean height and standard

deviation of each column in a picture were extracted from these ten measurements to mimic the features used for machine learning. The correlation between these ten images was then compared to the correlation between two approved but different cores and between an unapproved and unapproved core. To verify that the correlation between the same core is kept higher than between the other cores, implying that the readings are similar for the repeated measurements on the same core. To calculate the correlation, *scipy.stats.spearmanr* was used in Python.

To ensure that the accuracy of the sensor, *TriSpector1008*, seen in the datasheet [4] is sufficient to accurately measure the paper core damages a basic evaluation of the measured damages is done.

The size of the damages is calculated by cropping damages from various paper cores. The real-world equivalent values for width (*X-axis*) are calculated through *Equ. (3.1)* on a row. The row marked as cross-section B-B in *Figure 7*, seen in *Figure 9*, illustrates and sets the interval of columns used to calculate the actual width. The black lines, start and finish of the interval, draw the line between damage and paper through peaks of depth and height (Z-axis) in accordance with the cropping seen in *Figure 7*. The height of the damaged area (*Y-axis*) is calculated using *Equ. (3.2)* on a column. The column marked as cross-section A-A in *Figure 7*, shown in *Figure 9*, illustrates the column index interval used in the calculation, marked with black lines. Discernment between a deep mark and pressed paper is also done to generate two dimensions of the damages in the *Z-axis*, depth and height. A deep mark is present in the lighter upper part of the damage, encapsulated in the green box *Figure 7*, and pressed paper in the darker part. Depth (Z-axis) and height (Z-axis) are calculated through a mean value of the dark and white area of damage. This creates a span of millimeters in the dimensions $width * height * (Z_{depth}, Z_{height})$ with values comparable to and evaluated against the *TriSpector1008* height resolution.

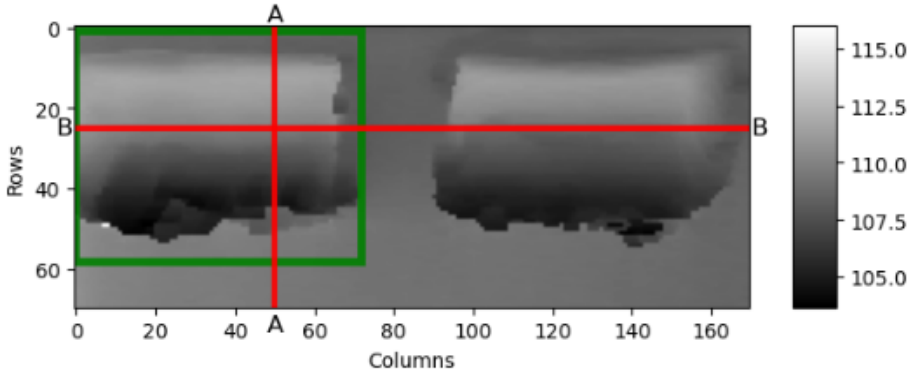


Figure 7. Cropping of chuck damage, green area marked for a visual representation of damage area. Cross-section A-A (red) = column 50, Cross-section B-B (red) = row 25. Roll 10, unapproved

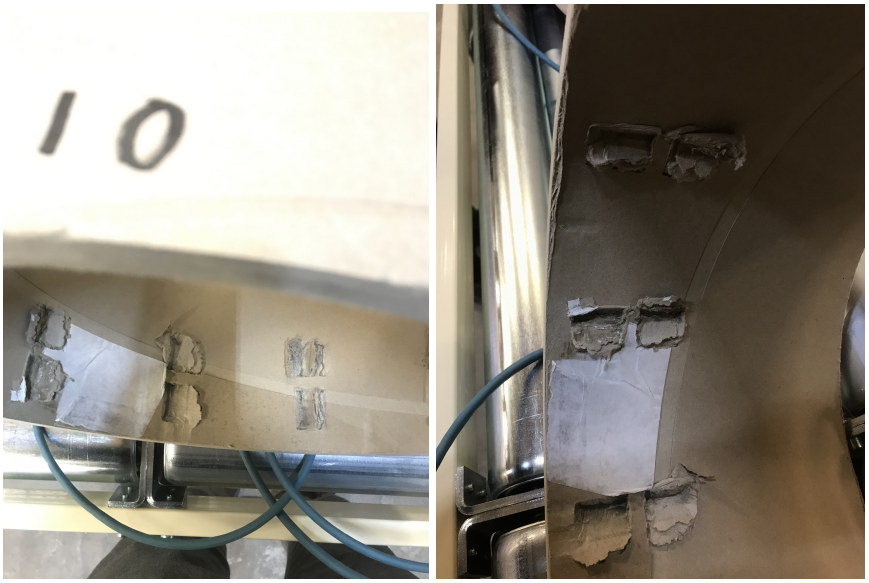


Figure 8. Photograph of Roll 10, unapproved and showing significant damage.

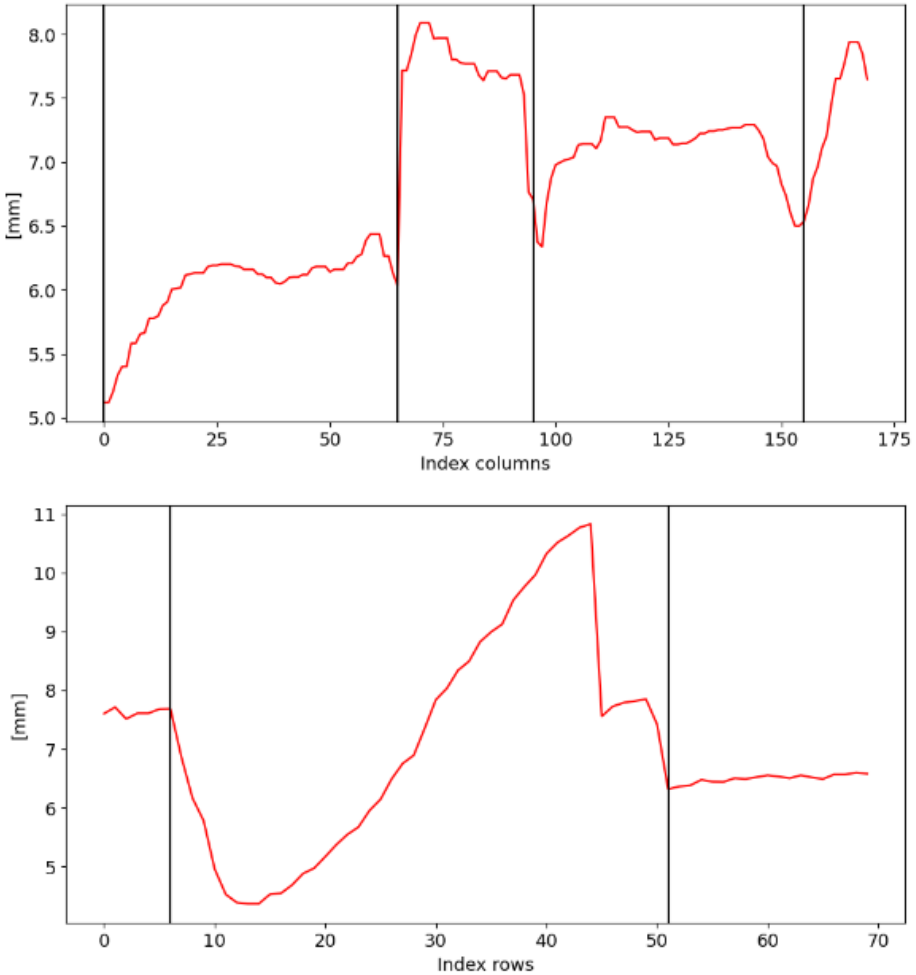


Figure 9. Plots of depth (Z-axis) on the column A-A (bottom plot) and the row B-B(top plot) of the cropping, in Figure 7. Black lines represent discernment between damaged and undamaged paper.

Baseline system

To calculate the threshold values for the system using a loop, all the mean, maximum and minimum values of each column in each picture were extracted. Using a training dataset, threshold values were manually chosen within these value spans and tested until a *false positive rate* of zero was achieved, as seen in *Figure 10*.

In order to mimic the threshold value system used by SOPAS, all of the pixels in each picture are looped through. If any pixel is outside the two threshold values, the paper core is labeled as unapproved. If all pixels are within the threshold values, the paper core is labeled as approved.

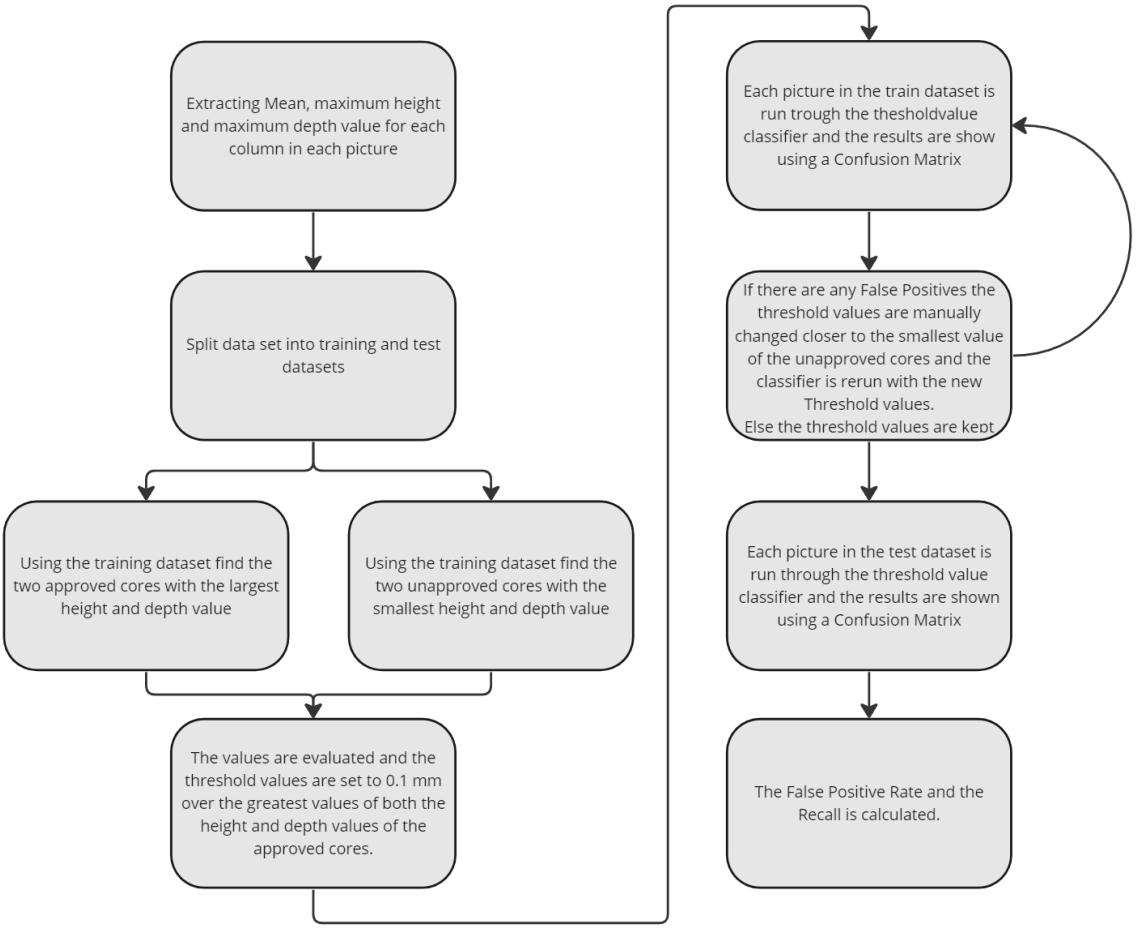


Figure 10. Process of choosing the threshold values.

3.7.2 Collecting dataset

When collecting the data, 27 different paper cores were provided with different damages. Each roll was labeled by the supplier of the paper cores. The paper core was loaded into the test rig, as seen in *Figure 11*, and then measured 20 times for each roll, each time with a different starting position. This created a dataset of 540 pictures, 280 labeled approved and 260 labeled

unapproved. The settings for the TriSpector1008 were set in SOPAS. The pixel resolution was set to 0.38mm per pixel in both x and y. The width was set to 74mm to cover the entire width of each paper core's damaged area. The length was set to 957.5mm, equal to the inside circumference of the paper core.

Figure 11. Test rig for reading the paper cores.

In the test rig, the roller wheel will run on the inside of the left side of the paper core, presented in *Figure* 12, which allows it to roll inside the core without coming in contact with the damaged areas of the paper core. The Trispector is positioned on the right side of the core enabling it to read the entire damaged area of the core, as in *Figure 12*

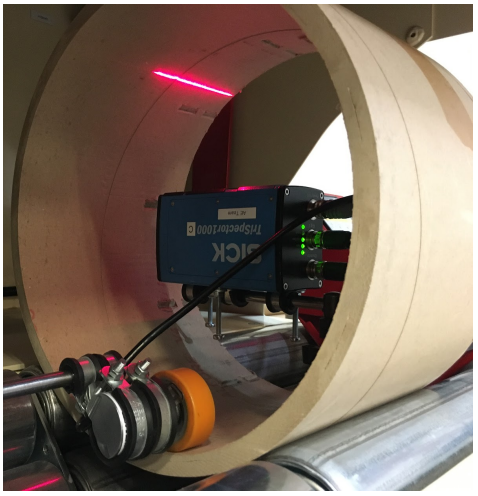


Figure 12. Presenting the roller wheel on the bottom left of the core and the TriSpector1008 in the middle right of the core. The TriSpector1008 is pointing upwards, reading the top of the core.

3.7.3 Image processing

Collected images are loaded from local disk using *Path* from *pathlib* and read using *imread* from *ImageIO.v3* simultaneously creating the labels by setting binary values, one or zero, corresponding to folder names ending

with *ok* or *not ok* using *fnmatch*. Images are split into intensity and depth by setting desired pixels to new arrays representing each part. The depth part is then decoded using the decoding protocol explained in *Chapter 3.4 Image Processing* seen in *Figure 13.*

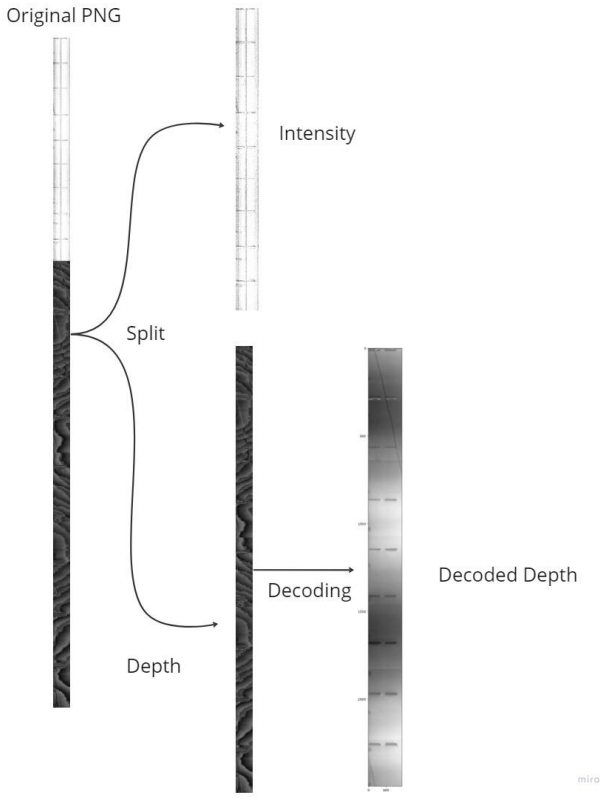


Figure 13 - Original PNG from a reading of Roll 9, Approved Paper Core

3.8.4 Feature extraction

Spatial domain

The depth (z-axis) of all pixels in a column is calculated through *Equ. (3.3)*. Maximum, minimum, mean and standard deviation are calculated through *numpy.max, numpy.min, numpy.mean* and *numpy.std* on the calculated depth. Classification learning algorithms K-nearest neighbor, Support Vector Machine, DecisionTrees and Random Forest were used in experiments on six feature spaces. The performance of the algorithms of each feature space is evaluated through confusion matrices with focus on the performance scores explained in *Chapter 3.6 Evaluating Algorithms*.

Table 3.1. Features f1 - f7 used in experiments on the spatial domain

Features	*f1*	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*
	max	*min*	$\bar{x}$	σ	$\bar{x} + \sigma$	$\bar{x} - \sigma$	$\bar{x}$ (**P**)

Table 3.2. Feature spaces 1-7 containing feature vectors of features from Table 3.1

Feature Space	1	2	3	4	5	6
column 1	*f1*	*f2*	*f3*	*f4*	*f7*	*f5 + f6*
column 2	*f1*	*f2*	*f3*	*f4*	*f7*	*f5 + f6*
column 3	*f1*	*f2*	*f3*	*f4*	*f7*	*f5 + f6*
column 5	*f1*	*f2*	*f3*	*f4*	*f7*	*f5 + f6*
column 6	*f1*	*f2*	*f3*	*f4*	*f7*	*f5 + f6*
column 7	*f1*	*f2*	*f3*	*f4*	*f7*	*f5 + f6*
....	.....	.....	.....	.....	.....	.....
column 192	*f1*	*f2*	*f3*	*f4*	*f7*	*f5 + f6*
Tot. features	192	192	192	192	192	384

The features *max*, *min*, *mean, standard deviation* and *mean* added/subtracted *standard deviation*, seen in *Table 3.1,* are calculated on each column of each decoded depth image creating a feature vector composed of 192 features in total. These feature vectors are either combined or kept separate creating feature spaces, a collection of n-dimensions of features, seen in *Table 3.2* essentially meaning a feature vector equaling 192 * $(nbr\ of\ features)$ features.

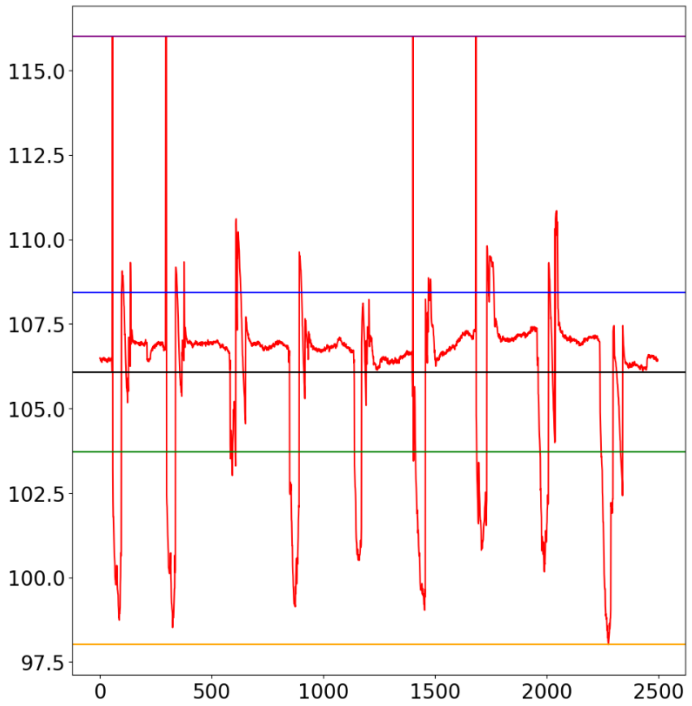


Figure 14. Example of features calculated on one column, plot of calculated depth for a column (red), mean (black), mean + standard deviation (blue), mean - standard deviation (green), maximum (purple), minimum (orange). Paper core 7 (approved), image number 155, column 150

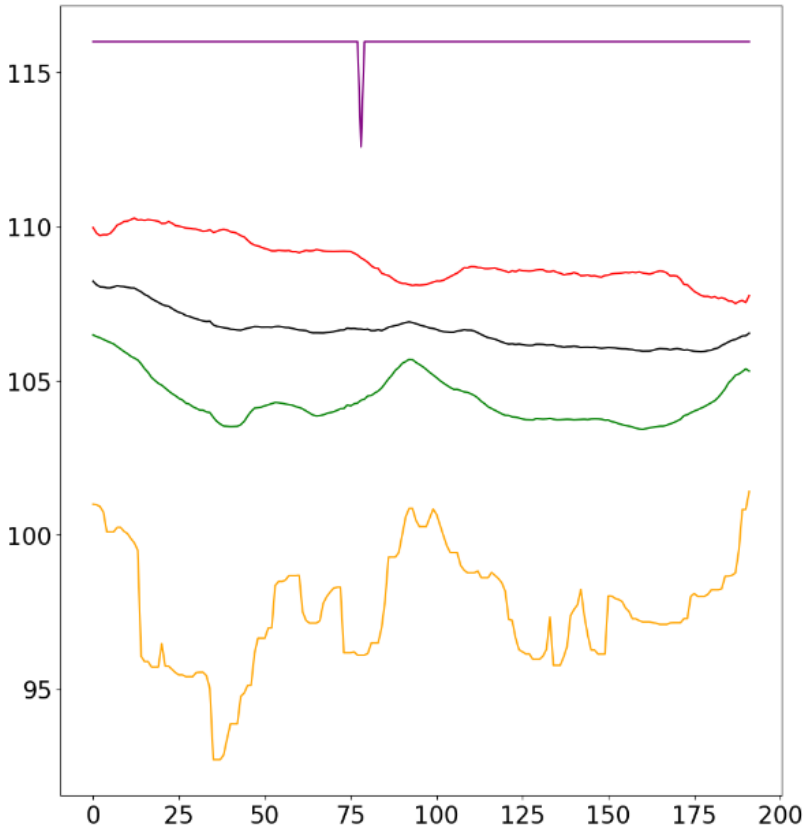


Figure 15. Example plot of mentioned features, except standard deviation, for all columns in one image. Mean (black), mean + standard deviation (blue), mean - standard deviation (green), maximum (purple), minimum (orange). Paper core 7 (approved), image number 155

From the experiments on features, mean value in itself is expected to not be adequate to classify the paper cores. Mean value is an estimate of all depth and height in a column which does not capture the magnitude of the peaks and bottoms of the damages adequately, as seen in *Figure 14*. Standard deviation measures the magnitude of peaks and bottoms or how far these are from the mean value. Mean, in this case, represents the height of the undamaged paper core surface. Standard deviation is expected to perform better than mean. Maximum and minimum are expected to perform the best since they capture the peaks and bottoms and emulate the depth graph better than the other statistical measures.

Frequency domain

The real Fast Fourier Transform of each column on all depth images are calculated through *rfft and rfftfreq* used in plotting of the Fourier Transform, both from *Scipy*. *Rfftfreq* calculates the sample frequency bin centers per unit of the sample spacing. *Figure 16* shows an example of a Fourier Transformed column. Feature *f7* from *Table 3.2* (mean value of power in all frequencies) of each column is extracted as a feature vector. Thus creating 192 features for each depth image seen in *Figure 17,* which shows both an approved and an unapproved paper core for comparison of these features.

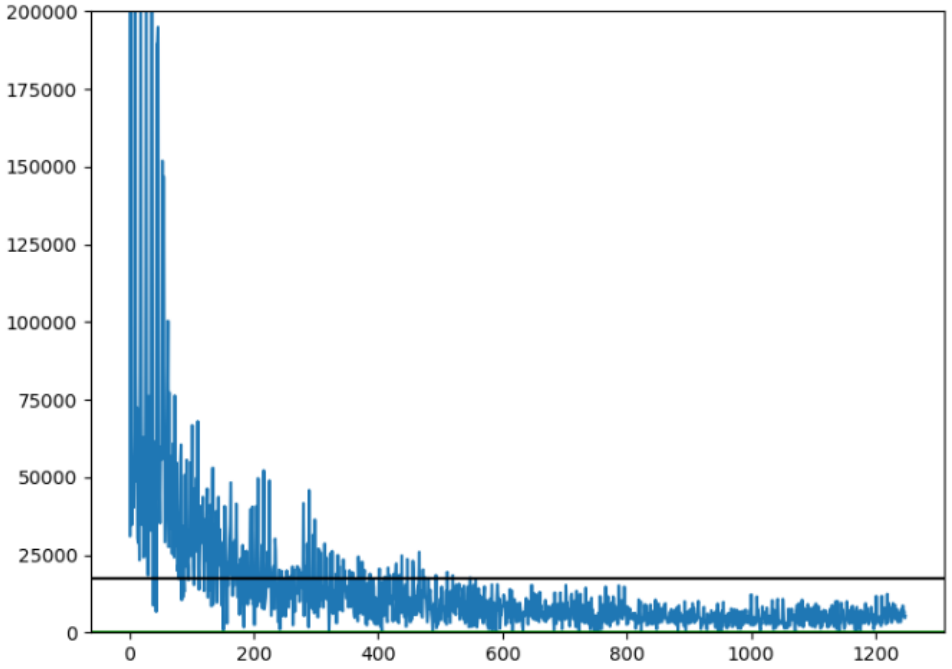


Figure 16. Example of f7 (mean) on a column, plot of a real Fourier Transformed column in a depth image (blue) and mean of power (black). Image number 500, column 130 paper core number 25 approved

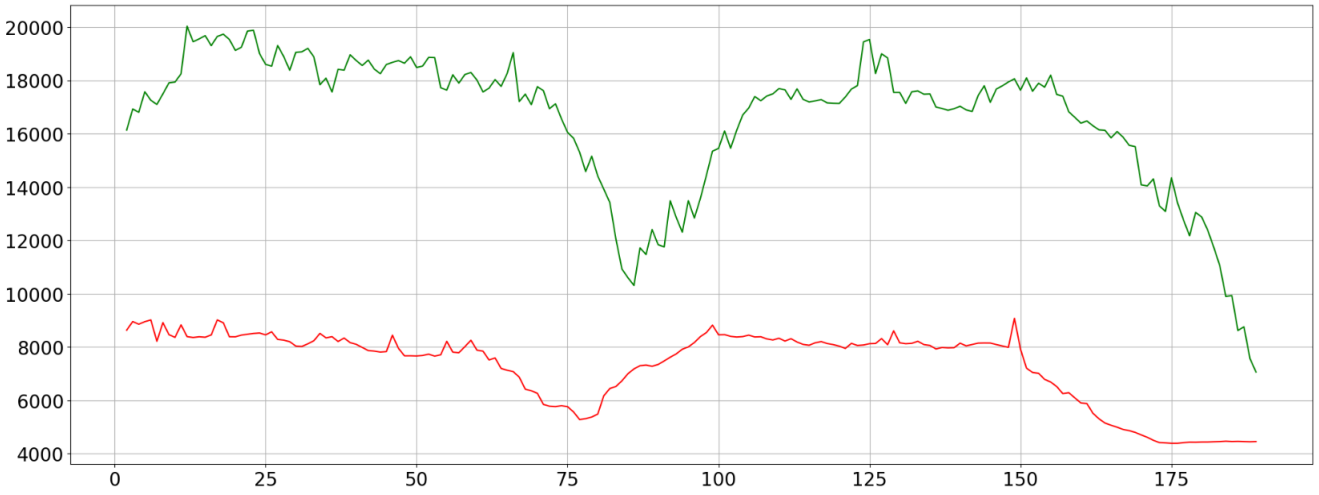


Figure 17. Example plot of the feature vector containing f7. Green = Approved, image number 201, paper core number 10. Red = Unapproved, image number 500, paper core number 25

Evaluation of Features

Feature spaces with the best performance Score will be evaluated using a Student's t-test. Performed on ten folds of a *StratifiedShuffleSplit* from *sklearn*, which is a stratified random cross-validation method with a modifiable test dataset size. Testing the robustness of the performance and equal performance results by having equal set sizes. This is done to prove that the performance of the features was not random and if there is a

difference between the performance of features or of a combination of features.

T-test is carried out through comparison of *recall* Equ. (3.6), calculating *sample mean* and *sample variance* of each comparison. These are used to calculate the t-statistic through Equ. (3.8), where *N* is the number of observations (k-folds). The null hypothesis is that the *recall* score of feature space A and feature space B are drawn from the same distribution, meaning there is no difference between the two feature spaces. The alternative hypothesis, the *recall* scores are drawn from two different distributions, meaning the features are actually different. One is better than the other.

$$t_{statistic} = \frac{\sqrt{N} * mean(|Recall(A) - Recall(B)|)}{variance(|Recall(A) - Recall(B)|)} \quad (3.8)$$

A t distribution table, seen in *Appendix [17]* with a degree of freedom *N-1* and significance level *p* of 95% is used to reject the null hypothesis of the t-statistic if it is outside the interval $[-t_{p/2, N-1}, t_{p/2, N-1}]$ [31].

3.7.5 Evaluation of algorithms

To verify that the *false positive rate* equals zero. A *false positive rate* scorer was created, taking the true labels and the predicted labels as input and returning the score calculated using *Equ. (3.5)*.

A *recall* scorer was created to ensure a high *recall* value is kept. The scorer takes the true labels and predicted labels as input and returns the *recall* score using *Equ. (3.6)*.

4. Results

4.1 Evaluation of current system

4.1.1 Sensors

The main component of the system is the TriSpector1008. To review the sensor, the sensing technique first had to be reviewed. This was done by comparing four techniques, Laser Triangulation, Stereoscopy, Moiré Interferometry and Holography.

Laser triangulation was found to be best suited for this case. Stereo has hardships with repeatability in setup changes, instability and freeform surfaces. Moiré amplifies small errors, enabling their detection. It has less computer time than laser triangulation during capture, but lacks in setup adjustment because of its complex design. Moirés very high precision is accompanied by a great acquisition cost. Holography performs better in precision and accuracy than triangulation but lacks speed and is sensitive to rig instability [15], which in an industrial environment could be horrendous.

The *TriSpector1008* from SICK and *In-sight 3DL4050* from Cognex were compared from a short evaluation and research of available market options for laser triangulation vision sensors. The comparison was made in accordance with key qualifications for industrial systems of robustness, dimensions as well as established and well-known brands.

Table 4.1. Model comparison of laser triangulation-based vision sensors see *Appendix [1-2]* and [4]

	TriSpector 1008	COGNEX In-sight 3DL4050
Size (laser pointing down)	136 mm x 62 mm x 84 mm	150.5 mm x 45mm x 101mm
Clearance Distance	56	92
Near FoV	40	55
Far FoV	75	90
Measurment Range	60	106
Heigth Resolution	20-50 µm	2,5-6,9µm
3d Resolution	0,049mm/px	0,0469mm/px
Points/Image	2500	1920

In-sight 3DL4050 was more precise, with a larger FoV (Field of View) but with a more significant clearance distance and of a larger size. This meant insertion into a paper core with a diameter of 12 inches would not be possible.

The repeatability of the sensor was tested by calculating the mean of the correlation between the ten measurements started from the same position. The *Spearman* correlation between each measurement was calculated and the mean of these calculations was equal to 0.98735474. Compared to two

approved cores, where the correlation was 0.6354623. One unapproved and one approved core compared gave 0.62457635, both seen in *Figure 18*.

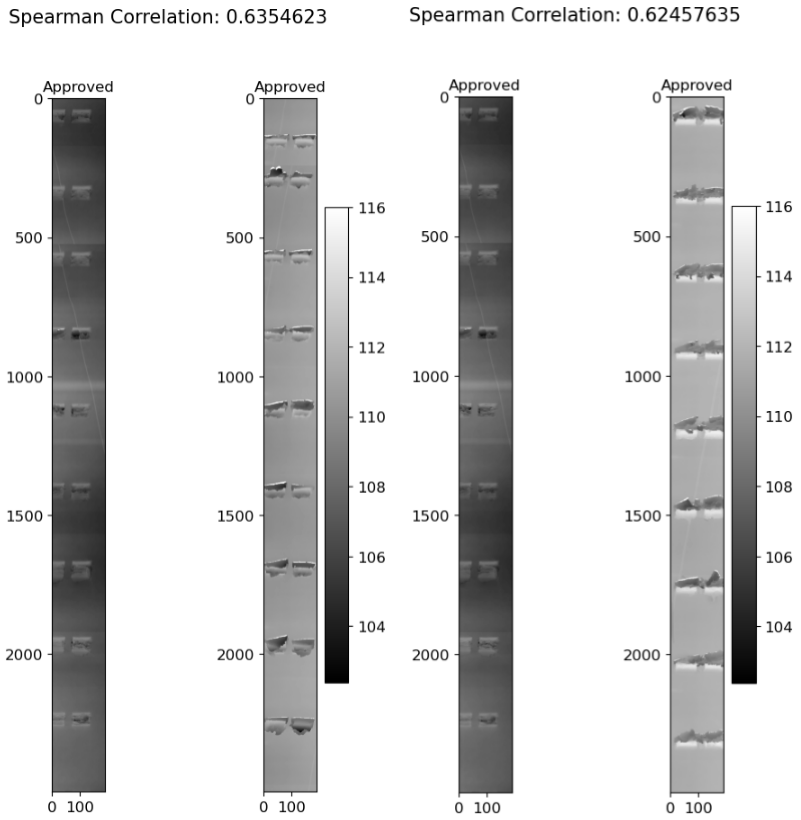


Figure 18. Correlation Between two approved cores on the left. Correlation between one approved and one unapproved on the right.

The damages measured were between 25 and 23 mm in width, 17 and 60 mm in length, 0.5 and 1.7 mm in depth, and 1.5 and 3.5 mm in height. All of the damages range on a millimeter scale, visual to the eye and the precision of the *TriSpector1008* is therefore more than enough.

4.1.2 Threshold system

When identifying the threshold values using the training dataset, the false positive rate equal to zero was achieved using the threshold values of 9.5mm from the mean of the core in depth and 10mm from the mean in height. This resulted in a *recall* value of 0.478. When the same threshold values were used on the test dataset, a *false positive rate* of zero was still achieved, but with a decline in recall value to 0.362. Both confusion matrices presenting the results can be seen in *Figure 19*.

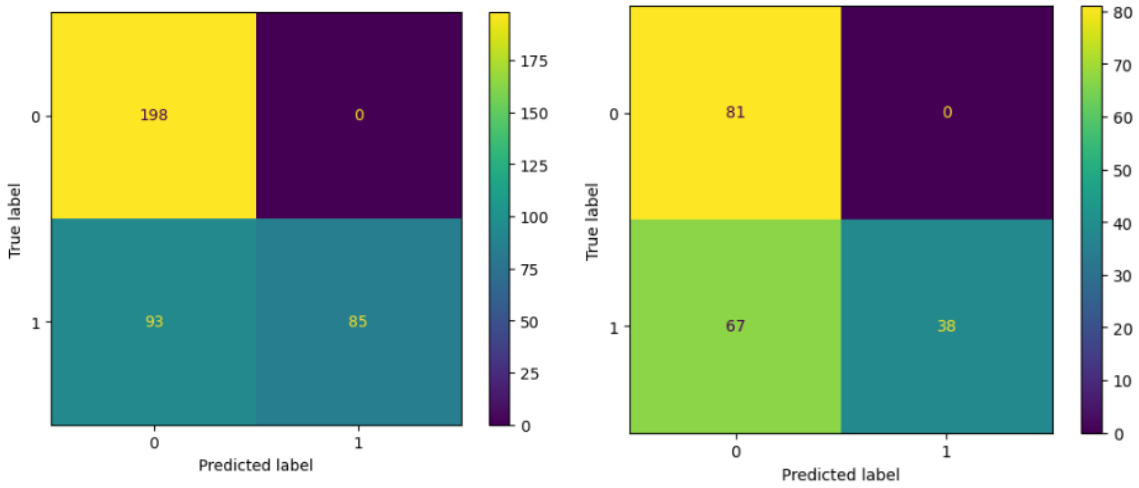


Figure 19. Confusion matrices showing the results of using the threshold values on the training dataset (left) and the test dataset (right) when achieving a False Positive Rate of zero

4.2 Collecting the dataset

When collecting the dataset, each of the 27 cores was photographed. In *Figure 20,* an approved core is shown and in *Figure 21,* an unapproved core is shown. 540 readings were gathered, 20 for each of the 27 cores and each with a different starting point in the scanning process.

Figure 20. Approved paper core. Showing small core. damages

Figure 21. Unapproved paper Showing significant damages.

4.3 Image processing

All readings (PNG-files) were processed according to the decoding protocol resulting in images like in *Appendix [19].*

4.4 Feature extraction

Student's t-test on the resulting feature space selections, done on *recall*, all achieved t-statistics values outside the interval, [-2.228, 2.228], thus rejecting the null-hypothesis of 5%, which is a commonly used threshold value for the p-value. The values of the *recall* score are therefore not random and support the alternative hypothesis that one feature space is better than the other.

4.5 Machine Learning models

Machine Learning model's performance of the five best feature spaces can be seen in *Table 4.5*. A table presenting all feature spaces and training scores is presented in *Appendix [16]*. Confusion matrices of the four algorithms on each feature can be seen in *Appendix [7-14]*. The results of a forward feature space selection based on *recall* scores and on ten folds of a random cross-validation method, can be seen in *Figure 22*.

Table 4.5. Performance plot of scores from the best feature spaces, Baseline and Machine Learning method. Green = best score, Red = worst score, Orange = Worst recall score, worst FPR score is within the worst score (red). Execution time is on training and in milliseconds.

Models	Score	Dataset				
Baseline						
	FPR	Test	0			
	Recall	Test	0,363			
	Exc Time	Train	0,483			
Models	Score	Dataset	feature space 2	feature space 3	feature space 4	feature space 6
Machine Learning						
K-NN	FPR	Test	**0**	0,026	**0**	**0**
	Recall	Test	**0,971**	0,951	**0,971**	**0,971**
	Exc Time	Train	0,608	0,581	0,545	1,032
SVM	FPR	Test	0,377	0	0,039	**0,701**
	Recall	Test	0,902	0,118	0,794	**1**
	Exc Time	Train	330,684	686,11	182,691	1282,43
Decision Trees	FPR	Test	0,039	0,104	0,039	0
	Recall	Test	0,843	0,902	0,912	0,961
	Exc Time	Train	30,627	27,342	23,982	44,617
Random Forest	FPR	Test	0	0,052	0,013	0,013
	Recall	Test	0,912	0,941	0,971	0,951
	Exc Time	Train	331,409	326,812	283,469	400,232

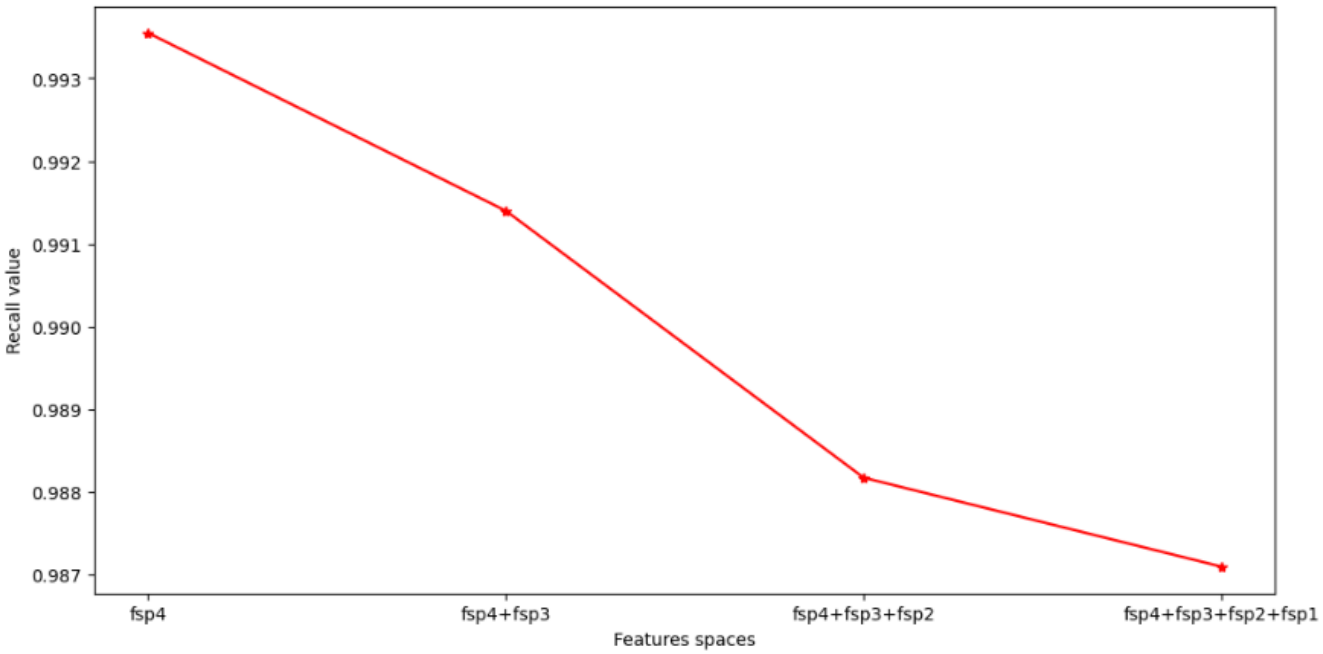


Figure 22. A plot of feature selection performed on feature spaces 1-4 based on recall score

5. Discussion

5.1 Collecting dataset

When collecting the dataset, too little caution surrounding the distance from the sensor and paper core was taken, leading to the depth measurement of chuck damages being maxed out in some rows in most readings. This can be seen in the feature *f1* (*max*) in *Figure 15*, where the maximum value (depth) is maxed out at 116 mm, which is the furthest distance of FOV the *TriSpector1008* can register. This meant that the information feature *f1* (*max*) could offer was limited and thereafter performed worse than expected but still relatively well.

5.2 Performance of Classification

When using Machine Learning, data is vital for a good system. The dataset used in this project could be considered small and could affect the result of the classification performance. The supplier prelabeled the cores used for training and testing using ocular inspection and evaluation from their professional opinion. Therefore a possibility of human error could be present, affecting the training and testing of the models and classification.

Since the baseline system was created in the project and not the actual one, it is possible that the baseline system created will outperform or underperform the current system's threshold methods.

When comparing the baseline system to the Machine Learning system, the Machine Learning system is clearly performing better, having a higher score on most of the algorithms, seen in *Appendix [16]*. With correct optimization, all algorithms would most likely have performed better than the baseline system. Baseline system had 67 false negatives, seen in *Figure 19*, versus Machine Learning and *f4* (*std*) which had 9 false negatives, seen in *Appendix 10*. Which means *f4* had 58 less unnecessary discardings roughly equating to 14 500 SEK saved when compared to the Baseline system.

The performance of the experiments on extracted features from the spatial domain gave all very similar results, as seen in *Appendix [16]* as well as *Chapter 4.5 Machine Learning models* and *Appendix [7-14]*. These results are substantiated by the PCA plots, seen in *Appendix [4-5]*, which from only two dimensions of the features *f2, f3, f4* (*min*, *mean, std*) reduced from the original 192 shows a majority of distinguishable clusters of approved and unapproved samples. Since features from the spatial domain lead to a better classification performance than expected, the theory surrounding insufficient spatial information is false and subsequent features extracted from the frequency domain are unnecessary and baseless. Student’s t-test

results in *Chapter 4.4* solidifies the performance score of features *f4* (*standard deviation*) in actuality, being robust and true. By cross-validating the dataset and achieving similar performance scores.

Since no similar classification problem was found on paper core damages it is hard to compare the results. However, the similar industrial classification problem mentioned in *Chapter 2.5 Machine Learning Classification in an Industrial Environment*, came to the same conclusion that classification using Machine Learning in industrial processes is capable of providing accurate and dependable classifications.

5.3 Societal demands

In today's society, more and more industries are turning their focus towards having a good environmental impact and a safe workplace where people are able to work without having to fear being injured. The project will help support both of these focus values while lowering the factory's cost.

Environmental

The project will decrease the number of paper cores wrongfully classified as unapproved and discarded. This will lower the number of paper cores used and the environmental impact by not having to create more paper cores than needed and instead reusing the ones as much as possible. This is in line with the Industrial Emissions Directive by the European Commission, which in 2022 revised the directive to increase focus on material efficiency and reuse [32].

Safety

The environmental impact will decrease while the industrial process safety is maintained since the *Score* requires the *false positive rate* to be zero, thus ensuring the system will not risk a paper core not approved for further use entering the system and possibly coming loose to harm the operator.

Economic

The project will also decrease the costs in production since it will reduce the number of paper cores discarded that would otherwise need to be replaced with new ones. A damaged core coming loose of the chuck would, in most cases, also increase the cost significantly as it would not only stop production but potentially damage machinery and personnel.

6. Conclusion

The project has successfully presented answers to the problem statements. It has identified the best features to differentiate an approved paper core from an unapproved one, analyzed the sensor used in the current system and proven the benefits of Machine Learning when classifying the paper cores compared to a baseline system.

Paper core approved for further use

What indicates an approved core was proven to be hard for a simpler system like the baseline system to identify. Rather, it was shown that a mathematical operation-based feature like standard deviation for each column was the best to identify whether the paper core was approved.

Current Sensors

When reviewing the sensor, the *TriSpector1008* from SICK was found to be the most suited for the task at hand. This was first proven by proving that the best-suited technique for measuring the damages on the cores in an industrial environment was Laser Triangulation. When compared with another laser triangulation sensor, the *TriSpector1008* proved to be the only one small enough to fit inside the 12-inch paper cores.

In order to validate that the sensor could deliver adequate measurements for the feature extraction, some damages on the core were measured, whereafter it was concluded that the *Trispector1008s* height resolution of 50 µm would be adequate to measure the damages of several millimeters.

To ensure the repeatability of the sensor, thus enabling the repeatability of the project, one paper core was measured ten times continuously, all starting from the same point when starting the reading. The features from these measurements were compared using *Spearman* correlation, where the mean of all ten measurements resulted in a correlation of 0.99. Whereas between two different approved paper cores, the correlation was only 0.64.

Machine Learning vs. Baseline System

The Machine Learning system proved better suited for classifying the reuse of paper cores with a large margin. The baseline system only had a performance score of 36.2 %, see *Table 4.5* compared to the Machine Learning system, using feature space four and the K-NN algorithm, a recall score of 99,14% was achieved when using random cross-validation using *StratifiedShuffleSplit.*

References

[1] A. Angelopoulos *et al.*, "Tackling Faults in the Industry 4.0 Era—A Survey of Machine-Learning Solutions and Key Aspects," *Sensors*, vol. 20, no. 1, Art. no. 1, Jan. 2020, doi: 10.3390/s20010109.

[2] C. Lee and C. Lim, "From technological development to social advance: A review of Industry 4.0 through machine learning," *Technol. Forecast. Soc. Change*, vol. 167, p. 120653, Jun. 2021, doi: 10.1016/j.techfore.2021.120653.

[3] "What are Paper Cores? | Western Container Corporation." https://www.westerncontainercorp.com/what-are-paper-cores.html (accessed Feb. 20, 2023).

[4] "V3T11S-MR12A7 | Vision | SICK." https://www.sick.com/se/sv/vision/3d-visionssensor/trispector1000/v3t11s-mr12a7/p/p448044?ff_data=JmZmX2lkPXA0NDgwNDQmZmZfbWFzdGVySWQ9cDQ0ODA0NCZmZl90aXRsZT1WM1QxMVMtTVIxMkE3JmZmX3F1ZXJ5PSZmZl9wb3M9NCZmZl9vcmlnUG9zPTQmZmZfcGFnZT0xJmZmX3BhZ2VTaXplPTI0JmZmX29yaWdQYWdlU2l6ZT0yNCZmZl9zaW1pPTkyLjA= (accessed Mar. 05, 2023).

[5] "TriSpector Plug-and-play 3D Vision Sensor | SICK." https://www.sick.com/gb/en/trispector-plug-and-play-3d-vision-sensor/w/press-TriSpector/ (accessed Feb. 24, 2023).

[6] "SOPAS Engineering Tool | SICK." https://www.sick.com/se/sv/sopas-engineering-tool/p/p367244 (accessed Mar. 05, 2023).

[7] "Vision Sensors and Their Modern Uses," *AZoSensors.com*, May 04, 2018. https://www.azosensors.com/article.aspx?ArticleID=1129 (accessed Mar. 06, 2023).

[8] "What Are Vision Sensors? | Sensor Basics: Introductory Guide to Sensors | KEYENCE." https://www.keyence.co.in/ss/products/sensor/sensorbasics/vision/info/#header (accessed Feb. 14, 2023).

[9] B. NT, "Machine vision systems: 1D, 2D, and 3D," *RoboticsBiz*, Jul. 07, 2021. https://roboticsbiz.com/machine-vision-systems-1d-2d-and-3d/ (accessed Mar. 06, 2023).

[10] "1D Vision Systems | Cognex." https://www.cognex.com/what-is/machine-vision/system-types/1d-vision (accessed Mar. 06, 2023).

[11] "3D Sensing - New Ways of Sensing the Environment," *FutureBridge*, Jul. 06, 2020. https://www.futurebridge.com/blog/3d-sensing-new-ways-of-sensing-the-environment/ (accessed Mar. 06, 2023).

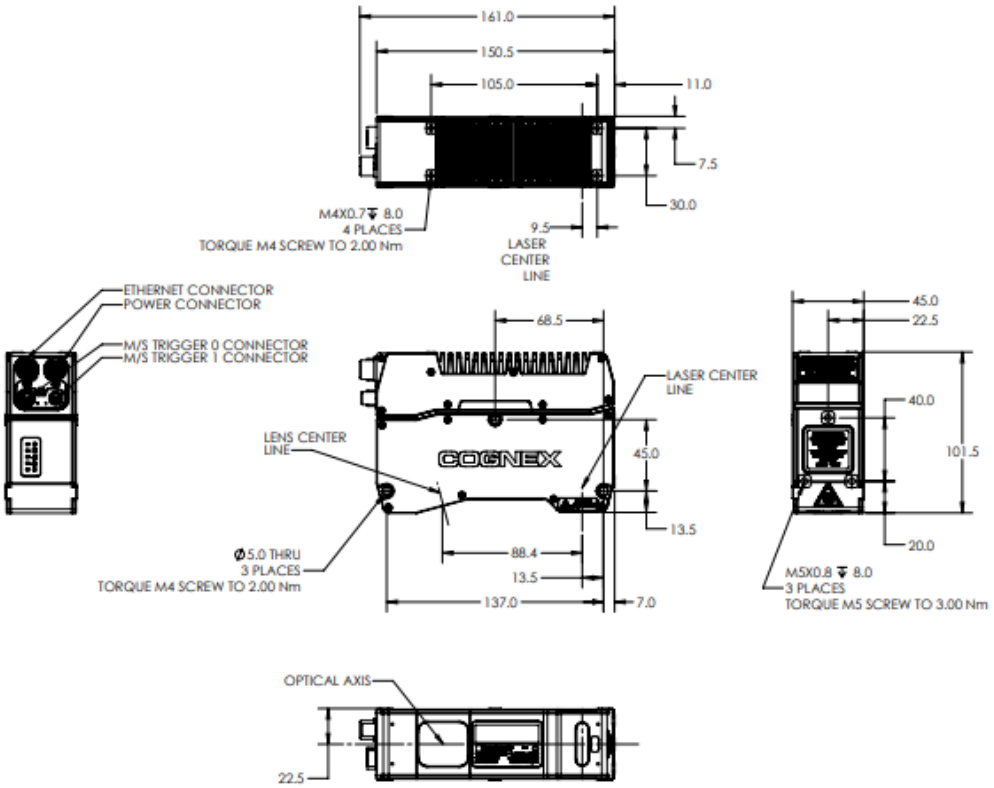
[12] G. Sansoni, M. Trebeschi, and F. Docchio, "State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation," *Sensors*, vol. 9, no. 1, pp. 568–601, 2009.

[13] "Measuring distance with light | Hamamatsu Photonics." https://hub.hamamatsu.com/us/en/application-notes/automotive/measuring-distance-with-light.html (accessed May 03, 2023).

[14] R. plc, “Renishaw: Interferometry explained,” *Renishaw*. http://www.renishaw.com/en/interferometry-explained--7854 (accessed May 03, 2023).
[15] “3D RECONSTRUCTION METHODS, A SURVEY:,” in *Proceedings of the First International Conference on Computer Vision Theory and Applications*, Setúbal, Portugal: SciTePress - Science and and Technology Publications, 2006, pp. 457–463. doi: 10.5220/0001369704570463.
[16] A. N. Thomsen, M. Kristiansen, E. Kristiansen, and B. Endelt, “Online measurement of the surface during laser forming,” *Int. J. Adv. Manuf. Technol.*, vol. 107, no. 3, pp. 1569–1579, Mar. 2020, doi: 10.1007/s00170-020-04950-6.
[17] T. R. Wanasinghe, B. R. Dowden, O. D. Silva, G. K. I. Mann, and C. Lundrigan, “Automated Seedling Height Assessment for Tree Nurseries Using Point Cloud Processing,” in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 3686–3691. doi: 10.1109/ICRA.2019.8793790.
[18] B. Charbonneau and B. B. Casper, “Wind tunnel tests inform Ammophila planting spacing for dune management,” Sep. 2018, Accessed: Mar. 02, 2023. [Online]. Available: https://eartharxiv.org/repository/view/1230/
[19] P. Magalhaes and N. Ferreira, “Inspection Application in an Industrial Environment with Collaborative Robots,” *Automation*, vol. 3, no. 2, Art. no. 2, Jun. 2022, doi: 10.3390/automation3020013.
[20] B. Mahesh, “Machine learning algorithms-a review,” *Int. J. Sci. Res. IJSRInternet*, vol. 9, pp. 381–386, 2020.
[21] N. Burkart and M. F. Huber, “A survey on the explainability of supervised machine learning,” *J. Artif. Intell. Res.*, vol. 70, pp. 245–317, 2021.
[22] K. K. Hiran, R. K. Jain, D. K. Lakhwani, and D. R. Doshi, *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition)*. BPB Publications, 2021.
[23] E. Alpaydin, *Introduction to Machine Learning, Fourth Edition*. Cambridge, UNITED STATES: MIT Press, 2020. Accessed: Feb. 28, 2023. [Online]. Available: http://ebookcentral.proquest.com/lib/halmstad/detail.action?docID=6676810
[24] R. Rai, M. K. Tiwari, D. Ivanov, and A. Dolgui, “Machine learning in manufacturing and industry 4.0 applications,” *Int. J. Prod. Res.*, vol. 59, no. 16, pp. 4773–4778, Aug. 2021, doi: 10.1080/00207543.2021.1956675.
[25] Y. Chen, H.-W. Ma, and G.-M. Zhang, “A support vector machine approach for classification of welding defects from ultrasonic signals,” *Nondestruct. Test. Eval.*, vol. 29, no. 3, pp. 243–254, Jul. 2014, doi: 10.1080/10589759.2014.914210.
[26] “DBS36E-BBCP02048 | Encoders | SICK.” https://www.sick.com/ca/en/encoders/incremental-encoders/dbs3650/dbs36e-bbcp02048/p/p325792 (accessed Mar. 05, 2023).
[27] J. Brownlee, “4 Types of Classification Tasks in Machine Learning,” *MachineLearningMastery.com*, Apr. 07, 2020.

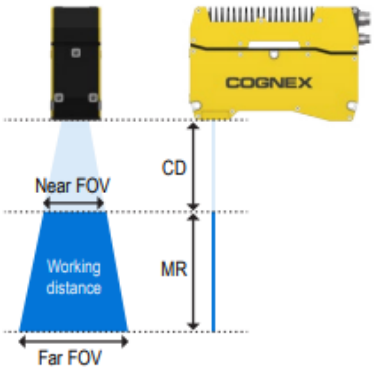
https://machinelearningmastery.com/types-of-classification-in-machine-learning/ (accessed Mar. 02, 2023).
[28] S. Ray, “A quick review of machine learning algorithms,” presented at the 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), IEEE, 2019, pp. 35–39.
[29] L. E. Peterson, “K-nearest neighbor,” *Scholarpedia*, vol. 4, no. 2, p. 1883, Feb. 2009, doi: 10.4249/scholarpedia.1883.
[30] A. Singh, N. Thakur, and A. Sharma, “A review of supervised machine learning algorithms,” presented at the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), Ieee, 2016, pp. 1310–1315.
[31] M. Politi, “Paired t-test to evaluate Machine Learning classifiers using Python,” *Medium*, Jul. 17, 2022. https://towardsdatascience.com/paired-t-test-to-evaluate-machine-learning-classifiers-1f395a6c93fa (accessed May 06, 2023).
[32] “Industrial Emissions Directive.” https://environment.ec.europa.eu/topics/industrial-emissions-and-accidents/industrial-emissions-directive_en (accessed May 03, 2023).

Appendix



In-Sight 3D-L4000 working distance

WORKING DISTANCE			
	IS3D-L4050	IS3D-L4100	IS3D-L4300
Clearance distance (CD)	92 mm	130 mm	180 mm
Near field of view	55 mm	75 mm	95 mm
Far field of view	90 mm	180 mm	460 mm
Measurement range (MR)	106 mm	235 mm	745 mm



Appendix 1. Data Sheet In-Sight 3D-L4000 3D Vision System, Cognex page 7

SPECIFICATIONS

			IS3D-L4050		IS3D-L4100		IS3D-L4300	
			1K	2K	1K	2K	1K	2K
Measurement range	Clearance distance		92.00 mm		130.00 mm		180.00 mm	
	Z-axis (height)	Measurement range	106.00 mm		235.00 mm		745.00 mm	
	X-axis (width)	Near field of view	55.00 mm		75.00 mm		95.00 mm	
		Middle field of view	72.50 mm		127.50 mm		277.50 mm	
		Far field of view	90.00 mm		180.00 mm		460.00 mm	
Laser (light source)	Wavelength		450 nm					
	Laser class		2M					
	Output power		45 mW					
Spot size (middle field of view)			110 µm		181 µm		240 µm	
Sensor	Data points/profile		960 points[1]	1920 points	960 points[1]	1920 points	960 points[1]	1920 points
	X resolution	Top	57.3 µm	28.6 µm	78.1 µm	39.1 µm	99.0 µm	49.5 µm
		Bottom	93.8 µm	46.9 µm	187.5 µm	93.8 µm	479.2 µm	239.6 µm
	Z resolution	Top	2.5 µm		4.4 µm		6.9 µm	
		Bottom	6.9 µm		25.9 µm		147.5 µm	
	Z repeatability[2]	Top	0.5 µm		1 µm		2 µm	
		Bottom	0.5 µm		1 µm		2 µm	
	Z linearity[3]		0.06% of full scale (F.S.)		0.04% of full scale (F.S.)		0.05% of full scale (F.S.)	
	Temperature characteristics		0.01% of F.S./°C[4]					
Environmental resistance	Housing protection		IP65					
	Operation temperature[5]		0–45 °C (32–113 °F)					
	Storage temperature		-20–70 °C (-4–158 °F)					
	Maximum humidity		20 to 80% (no condensation)					
	Vibration		10 to 57 Hz, double amplitude 1.5 mm X,Y,Z, 3 hours in each direction					
	Shock		15G/6 msec					
Scan rate			Up to 4 kHz (after windowing down the sensor) ([6]Up to 6 kHz)					
Housing material			Aluminum					
Weight			0.94 kg					
Dimensions			150.5 mm x 101 mm x 45 mm					
Power supply requirements			24 VDC +/- 10%, 750 mA minimum					
Inputs			Trigger, differential/single ended encoder, laser interlock					
Trigger			Input voltage limits: Trig+ - Trig - = -24 VDC to +24 VDC Input ON: >10 VDC (>6 mA) Input OFF: <2 VDC (<1.5 mA)					
Encoder specifications			Differential: A+/B+: 5–24V (1.0 MHz max) A-/B-: Inverted (A+/B+) Single ended: A+/B+: 12–24V (1.0 MHz max) A-/B-: VDC = ½ (A+/B+)					
Interface			Gigabit Ethernet interface Integrated link and traffic LEDs Standard M12-8 X-coded female connector					

1 Only available when binning is enabled in the camera settings.
2 Z repeatability is measured an average of 100 times over a pointcloud using a 4x4 mm area, at the middle of the measurement range.
3 Z linearity is the maximum deviation of 250 position measurements on the measurement range, where a measurement is the average of 2 profiles using the standard Cognex target.
4 From the standard ambient temperature 21 °C (70 °F).
5 Mounted to a 400 mm aluminum bar along the camera's fins (parallel to the camera) on top of the vision system.
6 When binning is enabled and the FOV is windowed down.

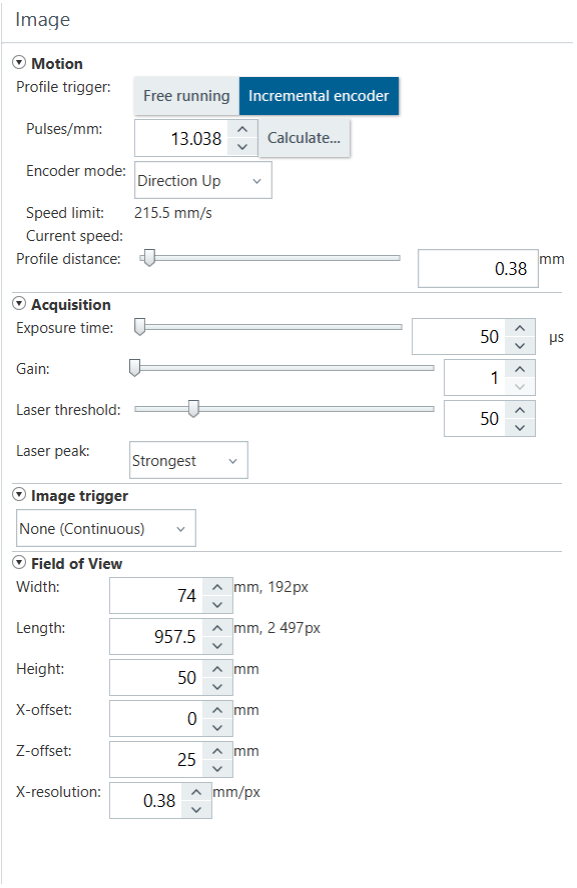
COGNEX

Companies around the world rely on Cognex vision and barcode reading solutions to optimize quality, drive down costs and control traceability.

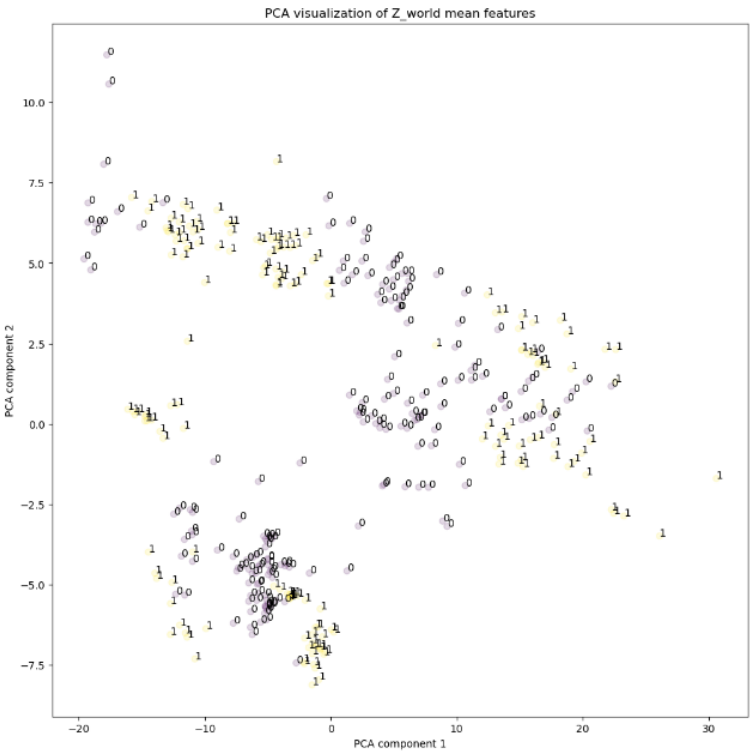
Corporate Headquarters **One Vision Drive Natick, MA 01760 USA** | For Regional Sales Offices, visit **www.cognex.com/sales**

www.cognex.com

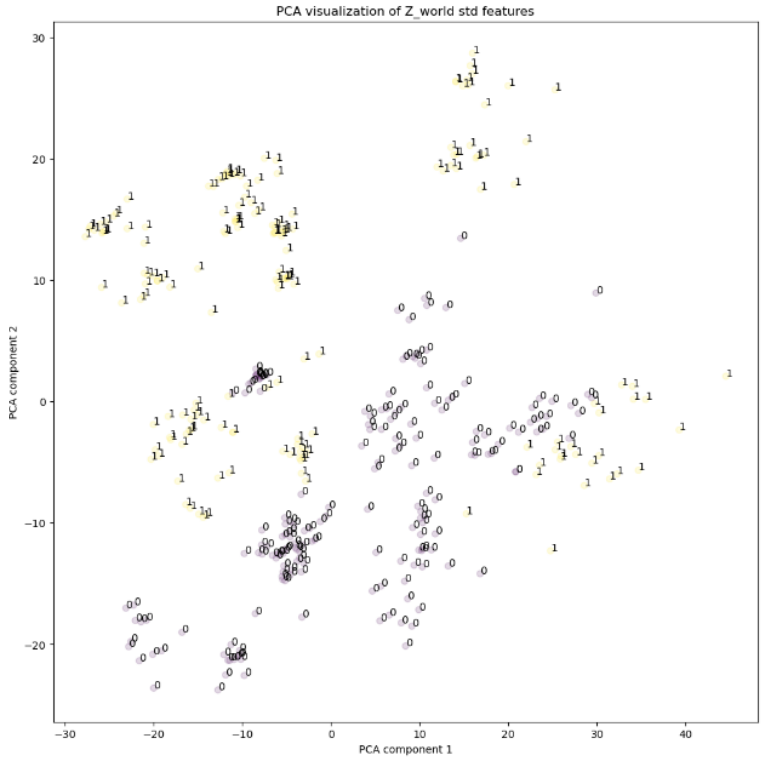
Appendix 2. Data Sheet In-Sight 3D-L4000 3D Vision System, Cognex page 8



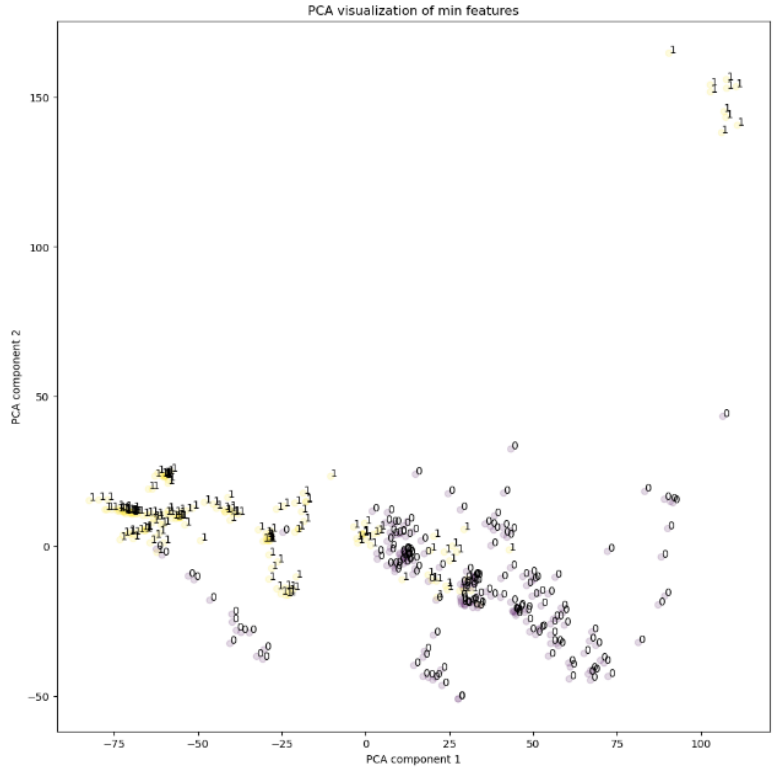
Appendix 3. Settings used for capturing readings of needed measurements, TriSpector1008 in SOPAS.



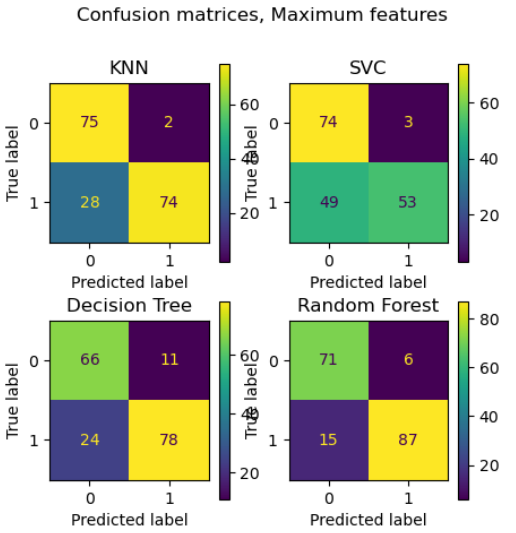
Appendix 4. PCA Visualization of Z-world mean features.



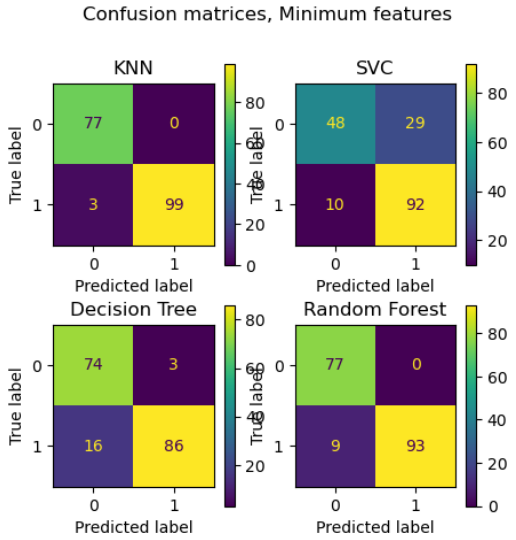
Appendix 5. PCA Visualization of Z-world standard deviation features.



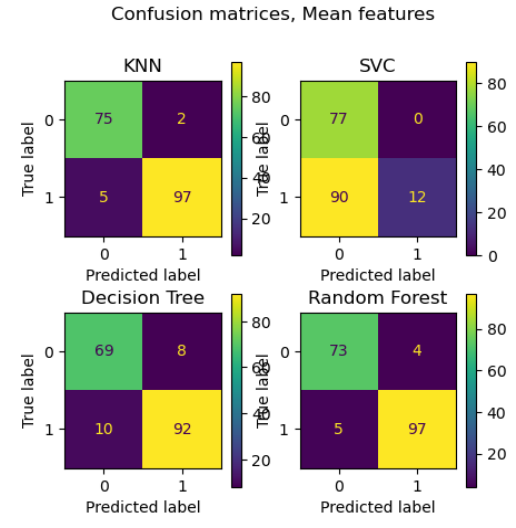
Appendix 6. PCA Visualization of Z-world minimum deviation features.



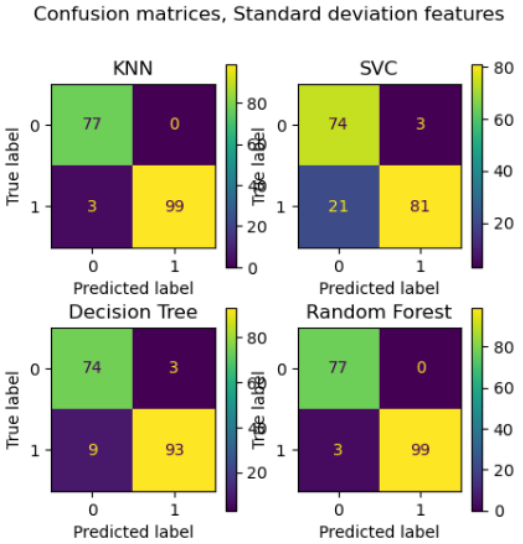
Appendix 7. Confusion Matrix of f1.



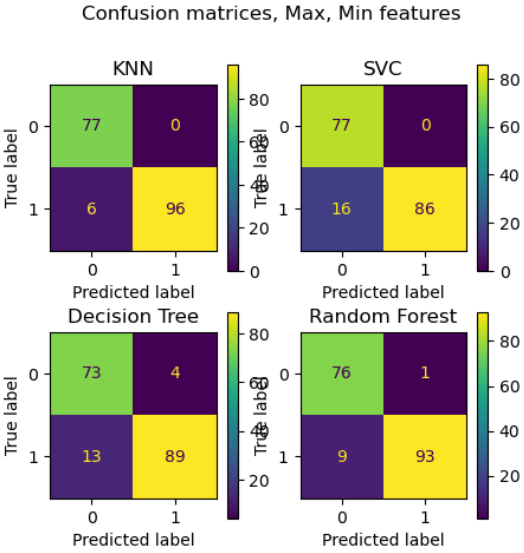
Appendix 8. Confusion Matrix of f2.



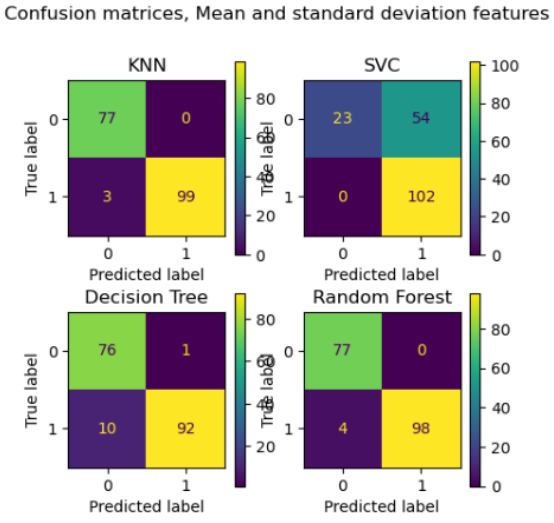
Appendix 9. Confusion Matrix of f3.



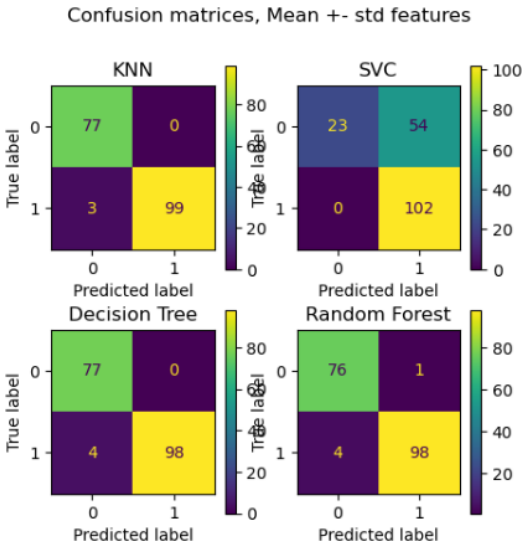
Appendix 10. Confusion Matrix of f4



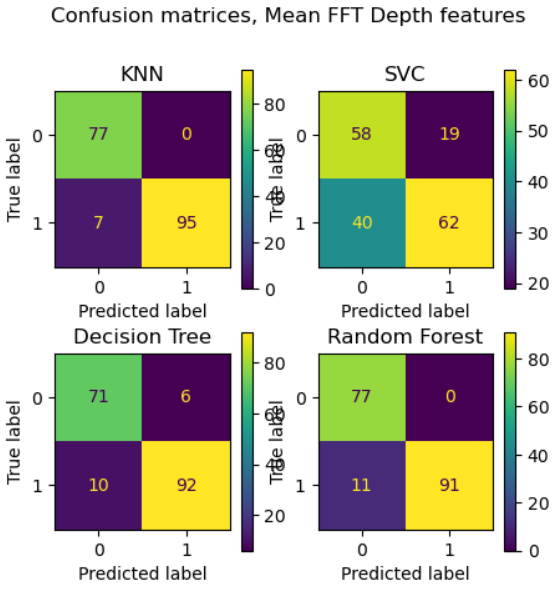
Appendix 11. Confusion Matrix of f1 + f2



Appendix 12. Confusion Matrix of f3 + f4



Appendix 13. Confusion Matrix of f5 + f6



Appendix 14. Confusion Matrix of feature Mean FFT depth.

Models	Score	Dataset						
Baseline								
	FPR	Train	0					
	FPR	Test	0					
	Recall	Train	0,478					
	Recall	Test	0,363					
	Exc Time		0,483					
Models	Score	Dataset	feature space 1	feature space 2	feature space 3	feature space 4	feature space 5	feature space 6
Machine Learning								
K-NN	FPR	Train	0	0	0	0	0	0
	FPR	Test	0,026	0	0,026	0	0	0
	Recall	Train	0,826	0,989	0,989	1	0,972	0,994
	Recall	Test	0,725	0,971	0,951	0,971	0,931	0,971
	Exc Time		0,802	0,608	0,581	0,545	0,707	1,032
SVM	FPR	Train	0	0,415	0,005	0,044	0,022	0,716
	FPR	Test	0,039	0,377	0	0,039	0,247	0,701
	Recall	Train	0,635	0,955	0,062	0,904	0,927	1
	Recall	Test	0,52	0,902	0,118	0,794	0,608	1
	Exc Time		655,981	330,684	686,11	182,691	770,469	1282,43
Decision Trees	FPR	Train	0	0	0	0	0	0
	FPR	Test	0,143	0,039	0,104	0,039	0,078	0
	Recall	Train	0,989	1	1	1	1	1
	Recall	Test	0,765	0,843	0,902	0,912	0,902	0,961
	Exc Time		10,922	30,627	27,342	23,982	38,324	44,617
Random Forest	FPR	Train	0	0	0	0	0	0
	FPR	Test	0,078	0	0,052	0,013	0	0,013
	Recall	Train	0,989	1	1	1	1	1
	Recall	Test	0,863	0,912	0,941	0,971	0,892	0,951
	Exc Time		236,215	331,409	326,812	283,469	328,202	400,232

Appendix 16. Full Table of Table 4.5. Performance plot of scores, Machine Learning method. Green = best score, Red = worst score, Orange = Worst recall score, worst FPR score is within the worst score (red). Execution time is done on training and is in milliseconds.

t Table

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

Appendix 17. T distribution table for Student t-test, t-statistic value

Conversion variables	Value
X0	-37
Y0	0
Z0	0
Dx	0,38743454
Dy	0,38349518
Dz	0,004545454

Appendix 18. Scalar values for conversion of reading (PNG-files) into real world depth values

Appendix 19. Example of a decoded depth image, Paper core 9