Assessment of standard definition data for high definition convolutional neural network for visual flaw detection

Juuso Varho

School of Science

Bachelor's thesis
Espoo 4.12.2023

Supervisor

Prof. Nuutti Hyvönen

Advisor

Prof. Iikka Virkkunen

Aalto University
School of Science

Aalto University
School of Science

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Tekniikan kandidaatintyön tiivistelmä

Tekijä Juuso Varho

Työn nimi Assessment of standard definition data for high definition convolutional neural network for visual flaw detection

Koulutusohjelma Teknistieteellinen kandidaattiohjelma

Pääaine Matematiikka ja systeemitieteet **Pääaineen koodi** SCI3029

Vastuuopettaja Prof. Nuutti Hyvönen

Työn ohjaaja Prof. Iikka Virkkunen

Päivämäärä 4.12.2023 **Sivumäärä** 26+1 **Kieli** Englanti

Tiivistelmä

Ydinvoimaloiden operatiivisen turvallisuuden takaamiseksi ohjesäännöt vaativat reaktoreiden visuaalisen tarkastamisen säännöllisin väliajoin. Tarkastus suoritetaan käymällä läpi reaktorin pintarakenteita kameran avulla. Aiemmin käytössä olevaan kamerayksikköön on kehitetty koneoppimismalli tunnistamaan kamerakuvasta vikoja. Käytössä oleva standardilaatuinen (SD) kamera tullaan päivittämään teräväpiirtolaatuiseen (HD) kameraan, minkä vuoksi kehitetty koneoppimismallin tulee pystyä käsittelemään HD-laatuista kuvaa.

Tutkimuksessa tarkastellaan HD-laatuisen koneoppimismallin kehittämistä ydinreaktoreiden visuaalisten vikojen havaitsemiseksi SD-laatuisen datan avulla. Visuaalisessa tarkastuksessa käytettävä kamera tullaan päivittämään HD-laatuisen, minkä vuoksi tarvitaan HD-kuvalla toimiva koneoppimismalli viantunnistukseen. Koska todenmukaista HD-dataa ei ole vielä saatavilla, koulutuksessa, testauksessa ja validoinnissa käytetään SD-dataa. SD-dataa muunnetaan alkuperäisestä koosta kahdella tavalla: kuvan suurentamisella ja reunojen täyttämisellä (engl. padding), jotta voidaan simuloida HD-laatuista dataa. Käytössä on u-net-niminen konvolutiivinen neuroverkko (engl. convolutional neural network), joka tuottaa segmentointikuvan alkuperäisestä kuvasta.

Tutkimuksessa kehitettiin neljä HD mallia, jotka oli kehitetty käyttäen eri datan kokomuutosmenetelmiä. Kehitettyjä malleja verrataan aiemmin kehitettyyn SD malliin. Vertailussa käytetään kahta mittaria: segmentointitarkkuutta, jota mitataan leikkaus yli yhdisteen -arvolla (engl. intersection over union) (IOU) ja F_1-arvolla, sekä tarkastussuoriutumista, jota mitataan vian havaitsemisprosentilla ja virhelöytöjen määrällä.

Tulokset osoittavat, että HD-mallit tuottavat tarkan segmentoinnin, mutta ovat herkempiä visuaalisille poikkeavuuksille kuvissa, mikä lisää virhelöytöjen vaikutusta. Lisäksi huomataan, että HD-mallien suorituskyky on vahvasti riippuvainen valitusta kokomuutosmenetelmästä. Segmentointitarkkuus on samankaltainen HD-malleissa ja vertailtavassa SD-mallissa, ja osa HD-malleista kykenee segmentoimaan viat paremmin kuin SD-vertailumalli. Huomataan, että HD-mallit eivät kuitenkaan ole yhtä tehokkaita vikojen tunnistamisessa, ja ne tuottavat enemmän virhelöytöjä. Tulokset ovat riippuvaisia käytetyistä testikuvista, ja tarkempien tulosten saaminen vaatii isomman ja HD-laatuisen testijoukon. Kokonaisuudessaan tulokset osoittavat,

että HD-mallit kykenevät tunnistamaan ja segmentoimaan vikoja myös SD-laatuisella harjoitusaineistolla.

Tutkimuksessa havaittiin, että SD-dataa on mahdollista käyttää datalähteenä kunnes HD-dataa on saatavilla. Lisäksi todetaan SD-datan tarjoavan mahdollisuuden lisätä datan määrää ja varianssia myös HD-datan saatavuuden jälkeen.

Avainsanat Koneoppiminen, U-Net, NDT, Automaattinen Visuaalinen Tarkastaminen

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the bachelor's thesis

Author Juuso Varho

Title Thesis template

Degree programme Bachelor's Programme in Science and Technology

Major Mathematics and Systems Sciences **Code of major** SCI3029

Teacher in charge Prof. Nuutti Hyvönen

Advisor Prof. Iikka Virkkunen

Date 4.12.2023 **Number of pages** 26+1 **Language** English

Abstract

To ensure the safe operation of nuclear power plants, visual inspection of the reactors must be made at regular intervals. The inspection requires scanning the surfaces of reactors with a camera unit. A machine learning model was developed for previously used standard definition (SD) camera unit. The camera unit will be changed to high-definition (HD) quality which requires the model to be able to process HD-quality data.

This study focuses on the use of SD data in developing a HD machine learning model for visual flaw detection in nuclear reactors. For training, testing and validation, SD data is used as HD data was not available during the time of this study. The original SD data is resized with two methods, upscaling and padding the original image, in an effort to simulate HD quality data. A convolutional neural network called u-net is used which produces a segmentation map of the original image.

Four HD models were developed with varying data resize methods, which were compared to a previously developed SD model which is used as a baseline. Two metrics are used for comparison, segmentation accuracy with intersection-over-union (IOU) and F_1-score, and inspection performance with detection percentage and false call rate.

The HD models were concluded to produce an accurate segmentation of the images but had a higher sensitivity to visual anomalies, which led to a higher number of false calls. The performance of HD models was noted to be highly dependent on the chosen image resize method. The segmentation accuracy of the HD models was very similar to the baseline SD, where the HD outperformed the baseline model in some metrics. However, the HD models were not as effective in identifying flaws and produced a higher false call effect. These results are dependent on the test images used and should be reproduced with larger and HD-sized test set for more accurate results. Overall the results indicated HD models' ability to identify and segment flaws even with SD-sized training data.

It was determined that SD data can serve as a viable substitute data source until HD data becomes accessible. Furthermore, it has the potential to offer additional data quantity and variance even after the availability of HD data.

Keywords Machine Learning, U-Net, NDT, Automatic Visual Inspection

Contents

1 Introduction

Nuclear power is one of the main energy production methods in the world. There are around 450 nuclear power plants in the world producing 10 % of the world's electricity Krikorian (2020). To ensure their operational safety, regulations require a visual inspection to be done at regular intervals. The frequency of these visual inspections is based on a variety of factors, such as the design and operating history of the plant, the criticality of the components, and the result of the previous inspection IAEA (2009).

The inspections are done using a color or black-and-white camera which is attached to a control unit. The inspector watches the camera feed to search for visual anomalies on the surface, which will be referred to as indications. Detailed accounting is made of prior indication findings, even if the finding is not relevant, such as scratch marks. Relevant indications, such as cracks, are inspected with particular diligence, and any changes to the indication are recorded.

The inspection requires the reactor to be in shutdown condition, which is why the inspections are done during the yearly maintenance. The inspection requires the inspector to scan specific surfaces of the reactor thoroughly, from multiple angles and with multiple lighting conditions, which is very time-consuming. Additionally, keeping the reactors non-operational is very expensive, which is why the inspection and maintenance have a tight schedule. The inspections are done with time pressure and often overnight. These factors may induce human-factor issues, which can decrease the probability of finding relevant indications. To combat this issue, Trueflaw Ltd has previously developed a machine learning model to analyze the camera feed and flag potential indications to the inspector in real-time.

This thesis is a part of Trueflaw Ltd's project with the Electric Power Research Institute (EPRI) to further develop this machine learning model to aid in the visual reactor inspection. The inspections have previously been done with a camera with standard-definition (SD) quality which usually means the images are sized 720x576 pixels. The aim is to develop a machine learning model for high-definition (HD) size video, which means the size of one image is 1280x720 or greater. The main challenge in this is to find suitable training data for the model, as recordings of previous inspections are only in SD quality. The aim of this project is to further develop and assess the performance of a machine learning model for visual inspections. This thesis focuses on examining the use of SD video to train a machine learning model to analyze HD sized video.

2 Background

Automation of defect detection has been used in some sectors for over 30 years. In the manufacturing industry, classic machine vision has been used to detect surface flaws since the 1980s, and it was capable of assessing the surface quality in terms of colour and texture and identifying defects Smith et al. (2021). These tasks required the inspected objects to be unchanging in a fixed location in a fixed

position with unchanging lighting conditions. This is achievable in many industrial manufacturing production processes, however, this kind of environmental structuring would be impossible in the nuclear plant inspection. The environment for nuclear plant inspections varies highly in terms of background, colors, distance to objects, and defect shapes and sizes. For these reasons, classic machine vision has not been able to be used. For a visual inspection task in nuclear power plants, the tool used needs to be able to reliably flag the indications from unseen images and at the same time, not flag the non-relevant indications. The requirements of this type of tool match the capabilities of convolutional neural networks (CNNs) and deep learning.

With advanced and more efficient models, the integration of machine learning into various NDT inspection methods has become increasingly prominent. Notably, machine learning techniques have been successfully applied to identify defects in ultrasonic data Virkkunen et al. (2021b). The resulting model was able to achieve a level of performance surpassing human inspectors, referred to as superhuman performance. This improvement in accuracy and efficiency is the overall motivation behind using machine learning in NDT inspections. However, for safety reasons, the NDT field is a heavily regulated industry and the integration of machine learning into NDT inspection requires a careful approach to align with established standards to ensure safety and reliability. Especially with nuclear power plant inspections, the tools and methods used must go through a rigorous inspections themselves and must be approved by the relevant authorities.

The European Network for Inspection Qualification (ENIQ) is a network focused on addressing the reliability and effectiveness of non-destructive testing (NDT) for nuclear power plants and one of the main contributors of today's global qualification guidelines for in-service inspections (ISI). ENIQ has published a recommended practice report concerning the qualification of NDT systems that make use of machine learning Virkkunen et al. (2021a). The report outlines the challenges and benefits of the use of machine learning. The report states machine learning to be well suited in all inspection techniques where data can be digitized. The report highlights that while the performance of inspectors exhibit variation, machine learning systems would provide highly repeatable and consistent results. The report also provides guidance on how to address the challenges and qualify an inspection system using machine learning.

One of the most influential milestones in using deep learning in computer vision processes was the development of AlexNet model Krizhevsky et al. (2012). The model won multiple image recognition contests and utilized the graphical processing units (GPUs) during training, which allowed for much faster training which in turn allowed for deeper model architecture. The AlexNet is a classic CNN architecture, which consists of five convolutional layers, two hidden fully-connected layers, and one fully-connected output. In the convolutional layers, a "filter" or "kernel", is applied to the image, which is a matrix of weights that slides over the original image values and calculates the sum of their element-wise product Dumoulin and Visin (2016). By changing the weights in the filter, different patterns, such as edges, can be detected. The resulting image after this convolution process is then max pooled. This is done to reduce the spatial size of the output while retaining the most important information.

After repeating this process for five times, the resulting values were run through a fully connected layer, which is typically a product with a weight matrix, to produce a vector output.

The usage of GPUs for training and the network design of using convolutional layers followed by max pooling layers is still the core basis of current deep learning models. The rapid development of GPUs has enabled the development of larger and deeper models. In comparison, the AlexNet model had 60 million learnable parameters, as one of the best-performing image recognition models called BASIC-L has 7.5 billion parameters Chen et al. (2023).

In a typical convolution network, the network 'learns' by optimizing the weights in the convolution layers and in the fully connected layers. These types of models usually require a large number of training data samples and output only a vector of classification probabilities. However, in visual flaw detection, the number of samples is often very limited, and the desired output of a sample image should contain precise information about where the flaw is located. These same requirements apply to biomedical image segmentation, which is what the deep learning model u-net was developed for Ronneberger et al. (2015). The u-net model is fully convolutional, meaning it only consists of convolution and pooling layers. Removing the connected layers enables the model to produce a semantic segmentation image instead of a one-dimensional vector output. This, in our implementation, means the output will be the same size as the original image, where the values of the pixels represent probabilities of that pixel in the original image belonging to the class of interest, i.e. being a flaw.

The u-net model consists of a contracting path and an expanding path. Similarly to the classic convolution model, the contracting path consists of convolution and max pooling layers that downsample the image. However, instead of flattening the output as in the classic CNN model, the resulting output is then upsampled in an expanding path, which is more or less symmetrical to the contracting path, to restore the original spatial dimensions of the image. The role of the contracting path is to extract the most important features and to reduce the spatial dimensions to improve computational efficiency. These operations make the model more efficient with the cost of information loss. The expanding path is then needed to restore the spatial dimension of the image. However, to properly restore the image to its original size, and to place the extracted features correctly, additional information is needed from the contracting path. This is provided with the model's 'skip connections' which concatenate the information of a downsampling layer to the corresponding upsampling layer. This enables the model to make use of higher level details and context which improves segmentation accuracy. The model architecture is illustrated in Figure 1.

In the downsampling, bottleneck, and upsampling phases, convolution and Rectified Linear Unit -activation (ReLU) are applied multiple times. In the first convolution layers, the extracted features are simple, such as edges or basic shapes, such as circles or squares. With multiple convolution operations, the network is able to build a hierarchy of more complex features by combining the information from previous convolution layers. Using ReLU-activation after a convolution layer improves the

performance of the model. This activation function applies non-linearityy to the result of the convolution.

2.1 Research question

The transition to HD-quality cameras is motivated by the potential for increased efficiency and accuracy. With HD quality, an image taken of an area of interest can be captured with more details. Fine details and better-defined edges enable more precise and accurate detection of details for both automated systems and human inspectors. The HD-quality image is also able to provide the same level of detail from an area of interest as an SD camera from farther away. This means the captured area is larger which can improve efficiency during inspections.

The focus of this study is to examine whether SD quality data can be used to train HD-sized model without performance loss. Performance is measured with different metrics to describe how well defects could be found from a set of test images. Previously developed SD model is used as a baseline for comparison. If no significant performance drop is observed, can the SD data be considered suitable for HD model training.

3 Methods

3.1 Model

The machine learning model used is based on a convolutional neural network for image segmentation called u-net Ronneberger et al. (2015). The model was chosen based on it being suitable for large image segmentation and performing well in similar segmentation tasks. Furthermore, changing only data and model size makes the results easier to compare to previously acquired results.

The architecture of the used model is presented in Figure 1. The encoding path of the model is the left side of the U-shape, which captures hierarchical features from the input image. Each convolution layer increases the network's ability to abstract and understand more complex features. The original image gets downsampled in the encoding process which then needs to be upsampled before the output. The upsampling is done in the decoder path which reconstructs the segmentation image based on the features learned in the encoder path. This reconstruction is enhanced with the skip connections, which are represented by arrows between the encoder and decoder paths.

The encoder path consists of 6 Convolution block, and max pool operations, one Bottleneck block, and 5 Concat, Convolution -block, 6 upsampling operations, and one Output block. The used model has a total of 707713 trainable parameters.

The model is implemented with Tensorflow Abadi et al. (2015), version 2.11.0. Convolutions, ReLU-activations, batch normalizations, max poolings and upsampling, are implemented respectively with functions Conv2D, Activation('ReLU'), BatchNormalization, MaxPool2D, and UpSampling2D from Tensorflows keras.layers library.

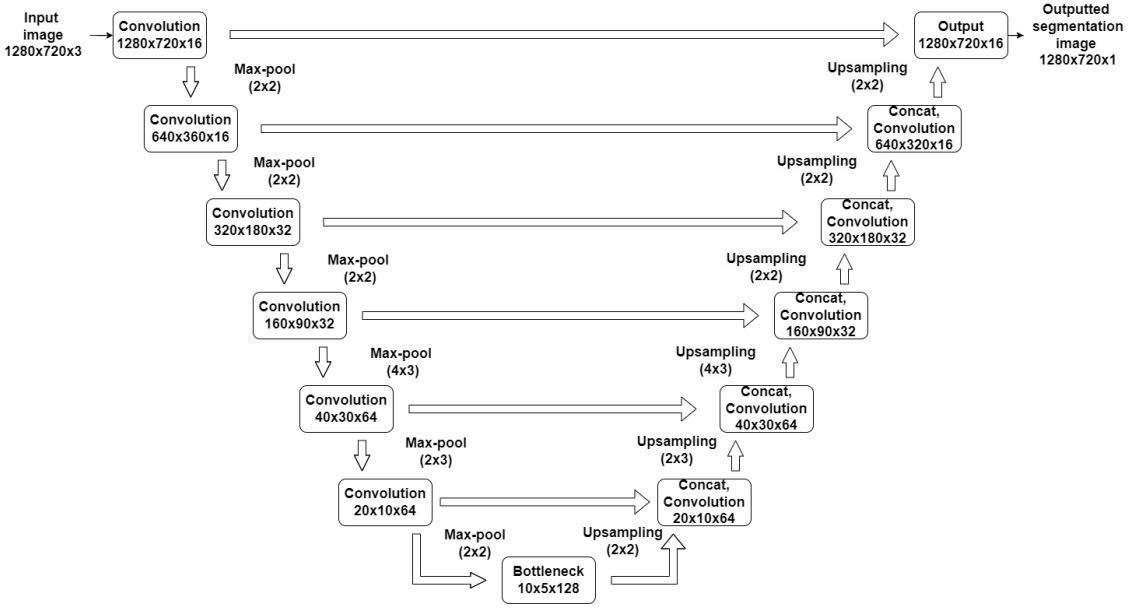

Figure 1: Implemented u-net architecture

In a Convolution block, convolution is applied to the image with a 3x3 convolution kernel with added zero-padding with ReLU -activation. This is repeated for a second time, with batch normalization applied before ReLU-activation. The resulting image has now only the non-negative values of the normalized distribution.

After one Convolution block, the image is max pooled with selected pool sizes, in which the pooled region is truncated to retain only the maximum value within the pool. This operation will decrease the spatial dimension of the image based on the size of the pool and stride size. The stride size determines step size between the pooling operations. The stride size used is the same as the pool size, i.e. there is no overlap in the pooled pixels.

After the encoder path, the image, which has dimensions 5x10, then goes through the Bottleneck block. Convolution and activation are repeated again two times, after which the image is upsampled with a specified size. This resizing uses the 'Nearest neighbor' method, which fills new pixels with the value of the nearest pixel. After this resizing, convolution and ReLU-activation are applied once more, after which the resulting image is outputted and will be taken as input in the decoder path. This Bottleneck phase is a crucial part of the network. The features learned in this block serve as a foundation for the subsequent layers to construct the segmentation map.

The second half of the network is the decoder path in which the image is expanded to its original size. The decoder path is symmetric to the encoder path. The decoder path consists of 5 'Concat, Convolution' blocks. In every block, the image of the previous block is first concatenated with the corresponding block from the encoder path, which is represented as an arrow in Figure 1. The resulting concatenated image then goes through similar operation as in the Convolution block, as we apply convolution, ReLu-activation, convolution, BatchNormalization, and ReLu-activation again. These are done to find more complex patterns in the concatenated image.

After these operations, the resulting image goes through the same upsampling process as after the bottleneck block, in which the image is resized and applied convolution and ReLU-activation.

The final step in the network is the Output block after which the resulting segmentation map is obtained. In the block, the input is concatenated with the corresponding Convolution block, and after this, convolution with 3x3 kernel and ReLU-activation is done twice. Now in the final step, convolution is done with the kernel size of 1x1 and with filter size 1 to produce 1 channel segmentation map. The activation function used in the last step is the Sigmoid -function. Using this activation function allows us to interpret the output of this as probability values. Typically, the predicted segmentation map is later thresholded to show values over 0.5 as one and everything else as zero. A successfully trained model should predict pixel values that are close to 0 or 1, which indicates a high confidence of segmentation.

3.2 Data

The used field data has been provided by EPRI and an undisclosed European nuclear power plant. The field data videos are recordings of the actual inspection, in which the inspector scans the reactor surfaces from different angles and in different lighting conditions. The field data is invaluable for the training of the model as it will be the environment in which the model will be used. All the inspection videos used are in SD size as HD-sized data is not available.

The amount of indication data is typically small compared to the amount of clean data available. As the indications can be diverse and complex, the model needs to have enough indication samples and variation to be able to learn and generalize the task. This means the training data set needs to have a diverse set of indications, of different sizes, shapes, and other relevant properties such as light reflection. In addition to indication representation, the training data set needs to contain enough of indication samples to not default to predicting the majority class. This, in the absence of enough indication data, means the model would consistently predict the absence of indications, which will result in a higher number of false negatives and lower sensitivity to detecting actual indications. To address this problem, laboratory samples and virtual flaws are added to the data set.

Virtual flaws are flaws that have been inserted into a 'clean sample', i.e. a sample without flaws. These virtual flaws are first extracted from their source sample and then added to a flaw dataset. The virtual flaws are inserted into an image with the eflaw function, which chooses a random flaw from the flaw dataset and inserts it into the image in a position where the flaw could naturally occur, such as an open surface or weld borders. For increased variety in the samples, the flaws are resized randomly before they are inserted into the sample. To make these virtual flaw samples look as realistic as possible, the inserted flaw is manipulated to match the conditions of the clean sample it will be inserted into. This process enables us to add variety and the quantity of the flawed data in the dataset, allowing us to use the limited amount of data more effectively.

To further increase the amount and variety of flaw samples and for the model,

Figure 2: Example laboratory flaw with the original SD image on the left and hand annotation on right. The image represents a typical flaw which might occur at weld borders.

laboratory samples are used in the data set. The laboratory data contains videos or images taken from samples that mimic the inspection environment. These laboratory samples contain indications that have been manufactured into them. These can be flaws, such as cracks, and non-flaws, including non-relevant indications like scratch marks. The non-relevant indications are added to help the model distinguish between between relevant and non-relevant indications. An example of laboratory produced sample is shown in Figure 2. The image exemplifies a flaw that can typically appear at the border of welds. For laboratory samples, the environmental conditions can be altered easily, such as the angle of the imaging, the angle and intensity of the light, and the distance between the camera and the sample. The real reactor environment is too complex to be modeled with laboratory samples. This is why they only work as additional source of data variety.

For large images, it is suggested in the original u-net paper Ronneberger et al. (2015) to use a tiling method for the image processing, in which the original image is divided into smaller tiles, which are then processed separately and later reattached. However, it was noted that dividing the image into smaller tiles makes it difficult for the model to predict flaws that are on the border region of the tiles. Furthermore, the processing speed of the final model is faster if the image can be inputted directly to the model. As the model is intended to work with the live feed of the camera unit in the inspection, this means the lag between the camera movements and the outputted video must be as small as possible. Dissecting every frame into tiles, running those tiles through the model, and combining the tiles takes additional processing power and time. The refresh rate in HD cameras is usually 24 frames-per-second (fps) but can also be even more. This would mean the model would have to process at least 24 images in a second. Using the described tiling method for each image takes longer than running a larger image through a larger model. The response time between the inspection unit and inspector needs to be fast enough to not be too noticeable as that could affect the performance of the inspector. It is for these reasons that the model was decided to process the whole image at once.

3.2.1 Data resizing

The main objective of this study is to establish the usability of SD images in HD model, and as the HD model can only be trained with HD-sized images, the SD data must be resized appropriately. This means choosing an appropriate resize method is a crucial step in this task as it directly impacts the quality of data.

Only one size of images can be used to train the model, which has been chosen to be of size 1280x720x3. This size was chosen based on it being the standard of high-definition resolutions, and the resolution can be changed in later model developments after the usability of the data resizing has been confirmed. The SD-sized data was resized in two ways, by upsampling the image itself, and by inserting padding to SD-sized images to increase the dimensions to HD-sized.

The padding method does not modify the original image but rather adds padding, meaning zero-pixels, to every side of the image to fit the dimension requirement. The frame placement, or the amount of padding on each side, was done randomly to change the frame position in each resulting image. This is done to prevent the model from learning any type of spatial preference in the occurrence of indications.

The other method of resizing is to use an upsampling function which adds pixels between the original pixels and uses interpolation to derive the new pixel values. This method was used to provide the model with larger images and indications. The training images are resized in two ways: resize them to as large as possible, and resize them to a random size between maximum and original size. Maximum resizing provides the model with the largest possible indications and the indications are proportionally the same size as for the SD model. Randomizing the size of the images adds variety to the sizes of indications in the dataset. This change in size also mimics the change in distance between the camera unit and the inspected surface.

In both resize methods, the aspect ratio, meaning the ratio between the image width and height, is preserved. As the aspect ratio of SD frames, $\frac{720}{480} = \frac{3}{2}$, is less than the HD frames, $\frac{1280}{720} = \frac{16}{9}$, some padding still needs to be added to the sides of the images resized with maximum upscale for them to be HD sized. Similarly, for the random resize method, padding is added to fit the size requirement of the model. For both methods, the amount of padding on each side is decided randomly for the same reasons as previously.

Upsampling the SD image to HD size results in a proportional increase in flaw size. It should therefore be as easy for the model to detect as for the SD model. This is done to provide the model with large indications which can span the width of the image. However, upsampling the image means new pixels are created with an interpolated value which increases the image size but worsens the image quality. Despite the selected interpolation function mitigating this effect, the quality does not match that of an original HD image. This drop in quality can make it harder for the model to detect the indications. Padding the original SD image does not change the quality, and the resulting image resembles a HD image with black borders. Employing both resizing methods enhances dataset variety and improves the model's generalization. Varying sizes of the images mimics varying the distance to the inspected surface.

Taking an SD image of an area of interest and padding it mimics taking a HD image of the same area but farther away. If the indications are found from the padded images, it means the HD model is able to find the same indications from a larger distance and a larger area of interest. With a larger imaging area, the inspections can be done more efficiently. If the distance to the surface is kept the same for the HD quality camera, it can capture the area of interest in more detail which improves indication detection.

The chosen upscaling method for the sample images is bicubic interpolation. It was chosen visually to provide accurate interpolation methods. In theory, bicubic interpolation works by considering the 4x4 pixel values surrounding the area of interest and applying a cubic spline to both horizontal and vertical dimensions of the 16 pixels. The new pixel value is calculated based on interpolating between the fitted splines. However, in practical applications, these operations truncate into a weighted sum of the 16 nearby pixel values which can be done efficiently with vector and matrix multiplications.

3.2.2 Data augmentation

Ronneberger et al. (2015) show that to efficiently make use of the limited amount of data, the data can be augmented in various ways to teach the network robustness and desired invariance. The flaws need to be recognized regardless of their position, orientation, and scale, and the model needs to be able to perform well even with noisy and imperfect data. Position and scale augmentation is done through image resizing and scaling virtual flaws. In addition to this, the data is augmented in three ways: image flipping, Gaussian noise, and Gaussian blurring. Every image is run through these augmentation functions which augments the images with some probability.

3.3 Development process

In this study, we build upon the foundation of the previously developed SD model to create the current code base. The earlier project served as the initial framework for our work, providing valuable insights and functionality that we extend and customize to meet the specific requirements of this project. First, without changes to the code, we train an SD model to work as a baseline for later models. In addition, this step confirms the functionality of the used code. After this step, the model is changed to work for HD-sized images. This requires changing the max pooling and upsampling dimensions to correspond with the dimension changes, and resizing the existing SD data to HD size. After verifying the model training is done correctly after these changes, the development is focused on optimizing the performance. This can be done by changing model parameters, for example, the number of convolution filters or layers, or changing the data augmentation methods, such as adding or decreasing the amount of Gaussian noise added. The optimize the performance, we need to establish methods of measuring it which is discussed in the next section.

After optimization, the final models used for this study can be trained. To learn how different resize methods affect the performance of the model, 4 different HD

models are developed. Three models, each with their own resize methods described previously, padding, max resize, and random resize, and a fourth model, which we call the 'full model', which combines all these three methods.

3.4 Performance evaluation

The used models output a segmentation map of the previous image with pixel-wise prediction of being a flaw. The prediction of one pixel can be categorized into four groups: true positive, false positive, true negative, and false negative. These categorizations are visualized in Table 1. Each annotated pixel is categorized into these four categories. When referring to these values, the referred value is the amount of these categories in one sample.

Table 1: Classes of classification outcomes

		Predicted condition	
		Positive	Negative
Real condition	Positive	True Positive (TP) annotated correctly	False Negative (TN) Real indication left unannotated
	Negative	False Positive (FP) Annotated but no real indication	True Negative (TN) correctly unannotated

For classification accuracy, typically precision and recall are used. Precision, also known as positive predictive value, means what percentage of the indicated values are correct. This is expressed in equation 1. Recall, also known as sensitivity, means what percentage of the real indication was annotated correctly. This is expressed in equation 2.

$$\text{Precision} = \frac{\text{correct annotations}}{\text{all annotations}} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

$$\text{Recall} = \frac{\text{correctly annotated pixels}}{\text{amount of pixels with real indication}} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2}$$

F_1-score is a measure of a model's performance which provides a balance between both the precision and the recall values. The F_1-score is expressed in 3 and returns the harmonic mean of the values.

$$\begin{aligned} F_1 = H(\text{precision}, \text{recall}) &= \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \\ &= \frac{2}{\frac{\text{TP}+\text{FP}}{\text{TP}} + \frac{\text{TP}+\text{FN}}{\text{TP}}} \\ &= \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \end{aligned} \tag{3}$$

The F_1 score is a good metric for calculating a balance between classification classes. However, it does not provide an understanding of the spatial overlap between the predicted and true labels. For this purpose, we use Intersection Over Union

-metric (IOU) which measures the ratio between the intersection and union of the true and predicted labels. This is expressed in equation 4 with true label T and predicted label P, and also with the binary classification classes.

The used model produces a segmentation mask with continuous values between 0 and 1 representing the pixel-wise probability of being a flaw. In order to use the F_1-score and IOU metrics, the prediction label needs to be converted into binary. This is done by thresholding the values to convert values above the threshold to 1, and rest to zero. Typically, a threshold value of 0.5 is chosen. In addition, the threshold value of 0.9 is used for the IOU metric. This is to measure the confidence of the predicted values. If the IOU(0.5) value is much larger than the IOU(0.9), we can determine that the confidence of the model in the predictions is low. And if the values are similar in size, we can determine the model to be predict the labels with large confidence.

$$\begin{aligned} \mathrm{IOU} &= \frac{|\mathrm{T} \cap \mathrm{P}|}{|\mathrm{T} \cup \mathrm{P}|} \\ &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}. \end{aligned} \tag{4}$$

These well-established metrics are good in expressing different types of performance in the model. However, they cannot be used as the loss function. The problem in these is that they need the output to be a binary value, 0 or 1, or hit or miss. Cross entropy is chosen to be the loss function in the model's training, as it is able to use the non-binary confidence values of the pixels.

Cross entropy is a well established and typically used metric for classification problems as it quantifies the difference between probability distributions, as explained in De Boer et al. (2005). In this case, the distributions are the predicted probabilities and the true probability distribution. During the model's training, the objective of the model is to minimize this loss function, which means minimizing the difference between the true labels and the predicted labels. Cross entropy can be used also for multiclass classification problems that have more than 2 categories. Our model only has 2 categories, flaw or no-flaw, which means we use binary cross entropy. Furthermore, as the flaws are typically very small compared to the background, we are most concerned about minimizing the loss in the flawed area, meaning we care more about finding the flaw rather than marking wrongly non-flaws. To achieve this imbalanced importance, we assign a loss weight to the minority class. Overall, this makes the chosen loss function be weighted binary cross entropy.

During a visual inspection, it is important for the model to be able to create an accurate segmentation of a defect. However, it is as important for the model to be able to find all the existing defects. The described metrics are useful in training the model and estimating the segmentation accuracy but do not explicitly state whether all defects were found from the images. For this, a new metric is created which considers the hit percentages, misses, false calls, and false call sizes. This new metric will be referred to as the NDT metric. The defects and predictions are grouped from the labels by considering connected True-value pixels. For every defect in an image, we define a defect to be found or 'hit' if the hand-annotate defect label

has an intersection with the prediction label. The hit percentage is the percentage of defects found from an image. Similarly, the defect is defined to be not found or 'miss' if no intersection with the prediction label exists. Additionally, we define a prediction to be a false call if a prediction group does not have an intersection with the hand-annotated true label. To estimate how large false calls are made, the average number of false call pixels is calculated per image. We use the product of the number and size of false calls to estimate their combined effect. These metrics are calculated for each image and the average values are calculated for each model.

3.4.1 Implementation

To evaluate the performance of the developed models, they need to be tested on an unseen test set. For this, we have chosen an inspection video which contains a large indication. The video is filmed in SD quality, size 720x480, which we upscale for the models by adding padding. With this method, the resulting images are most representative of HD quality images as no interpolation or resizing to the original image happens.

From this video, 33 frames are extracted, where 15 frames contain the indication and 18 frames that do not. The frames containing the indication are hand-annotated with a brush tool. The diameter of the brush tool needs to be large enough for the annotator to be able to smoothly trace the indication and for the indication to be fully inside the annotation brush. Furthermore, the diameter needs to be as small as possible to not also annotate the background. After testing different sizes, a diameter of 6 pixels was decided. However, the indications are mostly 1-3 pixels in diameter, which means at least half of the annotated pixels do not contain the indication. This difference can be seen in Figure 3, where we see the difference in the predicted label on the right and hand-annotated label in the center. This will drastically affect the performance metrics as the models are trained to only label the indication. The metrics are still useful for comparing the performance of the models and to see whether the performance of the HD models differs significantly from the performance of the original SD model. Only visible parts of the indication is annotated, i.e. parts where the indication is present but is not visible are not annotated.

The segmentation accuracy metrics, meaning precision, recall, IOU, and F_1 are calculated for the prediction label for the flawed images. For clean images, the label is inverted, meaning these metrics are calculated for the background. For the NDT metric, hits and misses are not presented for clean images as both would be zero for every clean image.

In order to simulate HD quality, the SD test images need to be resized. This is done with the two previously discussed methods: padding and maximum resize. For better performance comparison, the images given as input to the SD model need to correspond to the changes made for HD resize. With the maximum resize method, the original SD image is given as input without modifications as it is already the maximum size for SD size. For the padded method, the SD image is padded to be HD size. For the SD baseline model, the image is then downsampled with the nearest

neighbor method to be SD sized again. This is to ensure the input images are as similar as possible, as both then have black frames and similar relative indication size.

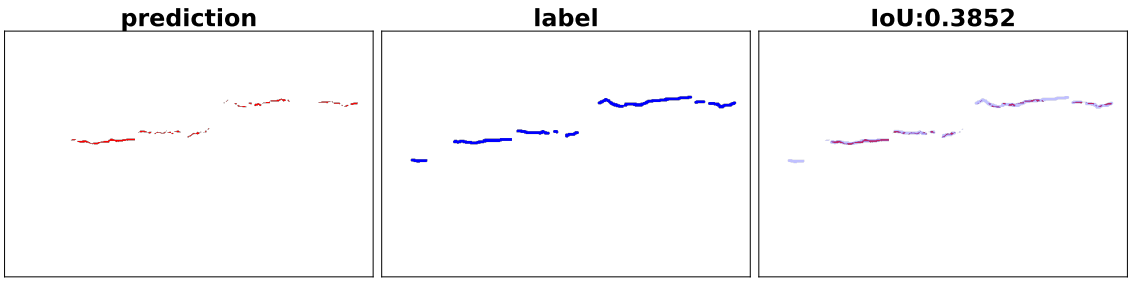

Figure 3: Visual comparison of model results and IOU metric. This consists of prediction label with threshold of 0.5, and the hand annotated true label. The IOU is plotted in the rightmost image, where the union is colored with light blue, and the intersections as red. The IOU50 score can be seen as the title of the third plot.

4 Results

Four different models were developed with the three different resize methods mentioned earlier. Each of the methods had multiple models developed with different model parameters, and only the best was used in comparing the results.

The used model had 708161 total parameters, of which 707713 are trainable. The training was run for 100 epochs consisting of 1000 steps. However, early stopping was used, which stops the training if the validation loss decreases for 30 consecutive epochs, and then it returns the best model in terms of validation loss.

- Precision 1
- Recall 2
- IOU(0.5) 4
- IOU(0.9) 4
- F_1-score 3

And for the NDT metric:

- Hit percentage
- Misses
- False calls
- False call size

- Product of the number of false calls and the size

The models are listed and named by model size and the resize method used. The produced models are either SD sized (720, 504, 1) or HD sized (1280, 720, 3).

0. SD model: original SD sized model developed to have a performance comparison for HD models.
1. Max resize model: HD sized model trained with SD images resized to fill the HD image.
2. Random resize model: HD sized model trained with SD images which have been resized randomly between the original and max resize size.
3. Padded only model: HD sized model trained with SD images which have not been resized, only padding has been added to
4. Full model: HD sized model which has been trained with SD images which have been resized with all available methods: max resize, random resize and padding in equal amounts.

Table 2: Segmentation metrics for flawed images with maximum resize. Threshold of 0.9 used for IOU90 and 0.5 otherwise

Model	Precision	Recall	IOU50	IOU90	F1
Full model	0.673	**0.436**	**0.323**	**0.215**	**0.445**
Padded only model	0.667	0.394	0.316	0.204	0.438
Random resize model	0.667	0.368	0.299	0.172	0.418
Max resize model	0.636	0.374	0.288	0.176	0.402
SD model	**0.830**	0.315	0.301	0.086	0.404

Table 3: Segmentation metrics for flawed test images with padding. Threshold of 0.9 used for IOU90 and 0.5 otherwise

Model	Precision	Recall	IOU50	IOU90	F1
Full model	0.640	**0.570**	0.390	0.280	0.530
Padded only model	0.690	0.530	0.390	**0.290**	0.520
Random resize model	0.670	0.500	0.370	0.240	0.500
Max resize model	0.700	0.460	0.330	0.210	0.460
SD model	**0.844**	0.496	**0.444**	0.061	**0.601**

The segmentation results are presented in Tables 2 and 3 for max resize and padded images, respectively, with the best values in each column in bold. With the max resize method, we can see that the 'Full model' achieves the best results in all

Table 4: NDT metrics for flawed images with maximum resize method and threshold value of 0.5

Model	Hit percentage	Misses	False call average amount	False call average size	Product of false call size and amount
Full model	0.61	3.13	4.40	8.66	38.09
Padded only model	0.64	2.93	6.73	5.64	37.97
Random resize model	0.52	3.00	1.53	7.72	11.84
Max resize model	0.50	4.00	**0.87**	7.07	**6.13**
SD model	**0.83**	**1.47**	3.87	**3.64**	14.06

Table 5: NDT metrics for flawed images with padded images and threshold value of 0.5

Model	Hit percentage	Misses	False call average amount	False call average size	Product of false call size and amount
Full model	0.784	1.733	10.867	8.277	89.939
Padded only model	0.795	1.600	14.667	14.736	216.127
Random resize model	0.574	2.467	4.067	4.932	20.056
Max resize model	0.580	3.467	**0.800**	**4.890**	**3.912**
SD model	**0.886**	**1.000**	3.467	8.253	28.612

Table 6: Segmentation and NDT metrics for clean test images with maximum resize. The segmentation accuracy is calculated for the background. Threshold of 0.9 used for IOU90 and 0.5 otherwise

Model	IOU50	IOU90	F1	False call average amount	False call average size	Product of false call size and amount
Full model	0.99965	0.99996	0.99982	11.79	82.48	972.29
Padded only model	0.99947	0.99982	0.99973	23.21	55.12	1279.47
Random resize model	0.99954	0.99986	0.99977	11.09	**35.17**	**390.08**
Max resize model	**0.99973**	**0.99996**	**0.99986**	**6.00**	74.84	449.01
SD model	0.9985	0.9997	0.9993	19.61	43.74	857.56

Table 7: Segmentation and NDT metrics for clean test images with padding. The segmentation accuracy is calculated for the background. Threshold of 0.9 used for IOU90 and 0.5 otherwise

Model	IOU50	IOU90	F1	False call average amount	False call average size	Product of false call size and amount
Full model	0.9981	0.9995	0.9991	18.12	47.09	853.38
Padded only model	0.9977	0.9991	0.9989	30.88	64.78	2000.3
Random resize model	0.9981	0.9993	0.9990	13.67	28.09	383.9
Max resize model	**0.9990**	0.9997	**0.9995**	**6.94**	30.88	**214.25**
SD model	0.9985	**0.9998**	0.9993	23.12	**22.80**	527.14

metrics but precision value, for which the SD model achieves the best value. For most of the categories, the segmentation accuracy values are very similar between the SD model and HD models. However, the baseline SD model has significantly lower IOU90 values compared to the HD models. The results obtained with padded test images in Table 3 show more variation in the best values. The overall segmentation accuracy values can be seen to be higher in almost every metric for every model. Despite the increased accuracy scores for the HD models, the baseline SD model achieves the highest precision, IOU50 and F_1 -scores. Overall, the HD models have

similar performances across all metrics, with the 'Full model' and the 'Padded only model' having the best IOU and F_1 scores.

In Tables 4 and 5, the results for NDT metrics are presented for max resize and padded images, respectively. With both methods, the baseline SD model achieves the highest hit percentage and the lowest amount of misses. Similarly to the segmentation results, the hit percentages of all models increase when the padding method is used. This is especially apparent with the 'Full model' and 'Padded only model' hit percentage scores which increased by around 25%. The performance of 'Full model' and 'Padded only model' seem to be very similar in terms of hit percentage and misses and get the best hit percentage scores among the HD models with both resize methods. However, these models seem to have a significantly bigger false call product value compared to other models. The 'Max resize model' and 'Random resize model' have both the lowest false call effect as the product of false call sizes and amounts is the lowest. However, the models produce the lowest hit percentage score with both test image resize methods.

The segmentation and NDT metrics are combined for the clean images and are presented in Tables 6 and 5 for max resize and padded resize methods, respectively. With the max resize method, we see the 'Max resize model' to produce the most accurate background segmentation and have the second lowest false call product score. The 'Random resize model' achieves the best and second best false call product scores for max resize and padding methods, respectively. The 'Full model' and 'Padded only model' can be seen to have the largest false call product values with both test image resize methods. The baseline SD model produces false call values somewhere between the highest and lowest values.

5 Discussion

From the numerical results, we see that the number of false calls made by the HD models seems to be correlated with the number of padded images provided, meaning smaller training data. This is most apparent with the 'Padded only model' consistently having the largest false call product values and the 'Max resize model' having the lowest. This is most likely due to padded images having smaller flaw sizes, meaning the model needs to have higher detection sensitivity compared to models trained with larger flaws present in training data. This would reflect the results acquired from the clean images presented in Tables 6 and 7, where the 'Max resize model' has the lowest false call product, but then has low hit percentage scores presented in Tables 4 and 5.

In both accuracy metrics, we see a trend of increased accuracy with the padded test images. This is likely due to the interpolation function with the max resize method affecting the image quality and sharpness. The flaws are typically darker than the surrounding area and have a sharp edge which helps to detect them. The image resize interpolates new pixel values between these flaw edge areas which makes the images appear smoother with the cost of decreasing image sharpness. The image sharpness caused by higher image resolution is one of the motives behind the

transition to HD-quality inspection cameras.

As can be seen from the segmentation accuracy values presented in Tables 2 and 3, the HD models achieve very similar segmentation accuracy scores as the baseline SD model. Moreover, as the IOU90 value is significantly higher, the segmentations are made with a higher confidence. Based on this, the results indicate HD models to be able to produce as accurate segmentations as an SD model.

In terms of NDT accuracy presented in Tables 4 and 5, the baseline SD achieves the highest hit percentage with a relatively low false call product. The differences between the hit percentages of SD and HD models is high with the max-sized test images, but become relatively similar between the 'Padded only model' and SD model when padded test images are used. The performance of the 'Full model' is very similar to the 'Padded only model' in terms of hit percentage and misses, but it produces fewer and smaller false calls with padded images. The decreased detection percentage with the max resized test images is likely again due to the decreased sharpness and quality, and the higher false call product in padded images is likely due to higher model sensitivity.

The objective of this study was to study the usability of SD-sized data in developing a HD-sized CNN model. The obtained results show that a HD model trained with SD images is able to achieve similar segmentation accuracy and indication detection percentage. The best results were achieved with the 'Full model' and 'Padded only model' with 'Full model' having fewer false calls. Based on this, SD images can be considered suitable for HD model training in the absence of true HD images, with the best results achieved with a model using all three types of discussed resize methods.

Limitation of these results is that they were obtained with a fairly small set of resized SD images. Most accurate results would have come from a large test set of HD-sized inspection data where the SD and HD images would be of the same area of interest. As this was not available, the used resized methods for test images were deemed to be sufficient for the purposes of this study.

For future models, a limiting factor with HD models can come from the increased memory requirements for model training. The size of the input image and the sizes of the proceeding convolution layers affect the required memory for model training. With larger images, the increased memory requirement can impact the depth of the model and the amount of model parameters, which can then affect model performance. As previously mentioned, the HD images can be divided into smaller sections with the tiling method, but this can introduce additional issues discussed previously.

With the confirmation of the usability of existing data, HD size can be used as a standard for future model sizes. The performance of later developed models can be improved even further, and these models can work as a foundation for models developed with actual HD-sized data. Furthermore, as the annotation of inspection video is labor intensive, and the amount of available data will be limited, existing SD data continues to be useful even after HD sized data is available. The benefits of using SD data in the HD models should be re-evaluated when enough HD data is available.

6 Conclusions

The focus of this study was to establish the usability of SD-sized data in training a HD-sized convolutional neural network for flaw detection in visual inspections. Four different HD models were developed which were compared to a baseline SD model developed in previous projects. The performance of the models was tested in terms of segmentation and inspection accuracy with multiple metrics. The performance of HD models was concluded to be similar to the used baseline SD model. For this reason, it was concluded that the use of SD data is appropriate for developing HD-sized models. This enables the use of vast amounts of existing inspection data to be used before true HD-sized inspection data becomes available, and also after to increase the amount of data and variety used for training.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms, 2023.

Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134:19–67, 2005.

Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

IAEA. *Integrity of Reactor Pressure Vessels in Nuclear Power Plants: Assessment of Irradiation Embrittlement Effects in Reactor Pressure Vessel Steels*. Number NP-T-3.11 in Nuclear Energy Series. INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2009. ISBN 978-92-0-101709-3.

Shant Krikorian. Preliminary nuclear power facts and figures for 2019, Jan 2020. URL https://www.iaea.org/newscenter/news/preliminary-nuclear-power-facts-and-figures-for-2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

Melvyn L. Smith, Lyndon N. Smith, and Mark F. Hansen. 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, 2021. ISSN 0166-3615. doi: https://doi.org/10.1016/j.compind.2021.103472. URL https://www.sciencedirect.com/science/article/pii/S0166361521000798.

Iikka Virkkunen, Martin Bolander, Heikki Myöhänen, Roberto Miorelli, Ola Johansson, Philip Kicherer, Chris Curtis, and Oliver Martin. Qualification of non-destructive testing systems that make use of machine learning: Eniq recommended practice 13. 2021a.

Iikka Virkkunen, Tuomas Koskinen, Oskari Jessen-Juhler, and Jari Rinta-Aho. Augmented ultrasonic data for machine learning. *Journal of Nondestructive Evaluation*, 40:1–11, 2021b.