

TRAINING A CAMERA BASED INSPECTION SYSTEM FOR APPEARANCE VARIABILITY

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

by
Vihang Yashodhan Khare
August 2020

Accepted by:
Dr. Adam Hoover, Committee Chair
Dr. Richard Brooks
Dr. Richard Groff

Abstract

This thesis considers the problem of variability in the appearance of machine parts while performing automated camera-based inspections in an appliance manufacturing plant. In an appliance manufacturing plant, machine parts are inspected to find any defects which might have occurred while assembling them. Training the system for the different appearances of these parts is important for detecting defects with high accuracy and precision. Machine parts have a lot of variability in their appearance and it is a difficult and time consuming task to train the inspection system for the same. Previously, a tool (clustering) was developed by our group that could automatically learn the different appearances of machine parts that needed inspection [7]. In this thesis, we perform different experiments using the tool with a goal of training an inspection system prototype for the learned variability. To do our experiments, we collected a total of 249,371 images for 9 inspection problems. A manual review of the data identified a range of 2-180 defects per problem. Our inspection system prototype, post training, achieved a range of 83% - 100% defect detection with a range of 0.0-2.9 false alarms per defect for 8 out of the 9 inspection problems. One of inspection problems was not firm in its structure, and in our experiments we could only find 50 % of the defects with a false alarms per defect rate of 5.0. Based on these results, we have designed a decision tree that could be used by engineers for training an inspection system for new inspection problems, for appearance variability.

Acknowledgments

I would like to thank my advisor, Dr. Adam Hoover and the entire department of Electrical and Computer Engineering at Clemson University for this wonderful opportunity. Working on this project has been the best learning experience of my life, both, personally and professionally. I feel it is a huge stepping stone to my career.

I would also like to thank the team at Samsung, especially Ravee Vaidhyanathan and Anatole Figueroa. Their guidance was critical for helping us collect and interpret data.

Table of Contents

Title Page	i
Abstract	ii
Acknowledgments	iii
List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Camera-based inspection system	2
1.2 Part appearance variability	4
1.3 Training for part appearance variability	5
1.4 Background	9
1.5 Related Work	24
1.6 Novelty	24
2 Methods	26
2.1 Data collection	26
2.2 Pre-processing	38
2.3 Ground truth	44
2.4 Clustering	45
2.5 Training and evaluation	47
3 Results	58
3.1 White water valve	58
3.2 Red water valve	61
3.3 Tub clamp hose	63
3.4 EMI Filter	66
3.5 Pressure Sensor	71
3.6 Wire holder	73
3.7 Thermistor screws	74
3.8 Screw near thermistor	77

3.9	Back screw	79
3.10	Training summary	80
4	Conclusion	81
	Appendices	88
A	Software	89
	Bibliography	90

List of Tables

2.1	Data collection summary.	28
2.2	Image pre-processing usage for different use cases depending on the strength of edges and lighting conditions.	42
2.3	Match methods used for different inspection problems.	43
2.4	Crops and ground truths for all the use cases.	46
2.5	A glimpse of the inspector output for the red water valve. The inspector stores the name of the image, maximum match score, the template that matched with that score and the ground truth label. In this case there were 4 templates numbered from 0 to 3.	52
2.6	A glimpse of the inspector output sorted in ascending order of the match scores for the red water valve for 4 templates, shown for the 1st least scoring part to the 500th least scoring part.	53
2.7	Values of different metrics, used for drawing the ROC curves for 4 red water valve training templates.	54
3.1	Inspection accuracies for all the uses cases post training.	80
4.1	Guide to choose RGB matching or Edge matching to find the location of the area of inspection.	84

List of Figures

1.1	A fill-level inspection. In Bottles 2 and 4, the fill-level of the beverage inside is below a threshold and hence they fail the inspection, while bottle 1 and 3 are full and they pass.	3
1.2	Screws inspected in the first three images match the three inspection templates well and pass the inspection. The fourth image has a missing screw and hence the inspection fails.	4
1.3	The screw in the fourth image fails the inspection because there is no inspection template that matches it well. The system erroneously stops the assembly line.	5
1.4	Variability in appearance due to change in viewing angle.	6
1.5	Variability in appearance due to incorrect assembling.	7
1.6	Variability in appearance due to change in color of background, the color of the part and specularities.	7
1.7	Variability in appearance due to change in orientation and occlusions.	8
1.8	Template matching using RGB images. The hose template matches with a score of 0.85 at the best match location (center of the part) as seen in the inspection window. A bright white spot can also be seen in the template match image on the right at the same location.	11
1.9	Template matching using edge images. The hose template matches with a score of 0.78 at the best match location (center of the part) as seen in the inspection window. A bright white spot can also be seen in the template match image on the right at the same location. The white spot in this case, is more distinct.	12
1.10	Overview of the inspection process.	13
1.11	Camera installation and a frame captured from the camera.	15
1.12	After locating the trigger template shown in a red box, the system locates different parts shown in green dotted boxes. The top left corner of the windows are at a fixed dx and dy from the trigger template. This is shown for the wire holder in this case.	16

1.13	A frame of the machine with different parts. After locating the image trigger, the system has defined an inspection window for the white water valve at a fixed distance from the image trigger. A match score is generated for the part (white water valve) inside the inspection window. The match score (0.85) is higher than a threshold and is hence labelled as an OK. The template match image of the inspection template and the window is also shown.	18
1.14	Overview of the cropping process.	19
1.15	The 40 cluster seeds capture different variations in the appearance of the screw including defects.	21
2.1	Overview of the methods used for our experiments.	27
2.2	Low Gain dataset: Six different parts inspected in 39268 such frames.	29
2.3	In the first figure, the white valve is present and fully secured and connected to the green mount (OK). Next, it is absent (NOK). In the third figure, it is dangling (NOK).	30
2.4	In the first figure, the red valve is present and fully secured and connected to the green mount (OK). Next it is absent (NOK). In the third figure, it is dangling (NOK).	31
2.5	In the first figure, the hose is seated on the tub and the clamp is secure (OK). Next, it is occluded by an object and it cannot be inspected (NOK).	31
2.6	In the first figure, the EMI filter is completely fastened down (OK). Next, it is not fastened down completely (NOK).	32
2.7	In the first figure, the white connector of the pressure sensor is snapped in completely inside the blue mount (OK). Next, the part is occluded or it is a trigger failure (NOK).	32
2.8	In the first figure, the wire harness is inside the wire holder (OK). Next, the wire holder is not tied and the wire is left dangling outside of the holder (NOK). Next, the holder is not tied and the wire can slip out (NOK). In the fourth figure, the wire is not going through the holder and in the last figure the holder is absent (NOKs).	33
2.9	Thermistor data set: A frame of the tub of the machine; Two different parts inspected in 23,247 such frames.	34
2.10	In the first figure, the thermistor frame is screwed into the tub completely (OK). In the second figure, there is a human hand covering the part and we cannot inspect it (NOK).	35
2.11	In the first figure, the screw is screwed tightly on to the frame (OK). In the second figure, the screw is absent (NOK).	35
2.12	Data Collection 3: Top left and top right corners of the back of the machine; Four screws at the top left corner and three screws at the top right corner inspected.	36

2.13	Screws on the back of the washing machine. The first figure is an OK screw. Next, the screw is absent (NOK). In the last figure, the screw is not screwed into the frame (NOK).	37
2.14	Different pre-processing options developed to improve the results of the clustering algorithm and other methods.	38
2.15	The system has an option to adjust the window size if the part is not fully visible in the cropped images.	39
2.16	Pre-processing drastically improves the edges of the parts which are dimly lit.	41
2.17	Pre-processing fails for this use case because the lighting is good and pre-processing is not required.	41
2.18	Process of manual reviewing for defects.	45
2.19	Overview of the methods used for training and evaluation of an inspection system prototype, for the learned variability.	48
2.20	The 40 cluster seeds capture different variations in the appearance of the EMI filter, including defects.	50
2.21	ROC curves of the red water valve for 1 training template and 4 training templates drawn for 1 least scoring part to 500 least scoring parts. For 4 training templates, the operating point is at the right bottom corner with a high defect detection rate and a low false alarm rate. The scores in this region are in the range of 0.58 to 0.62 and the threshold can be set to one of those values for inspecting this part.	55
3.1	The template chosen for the white valve.	58
3.2	OK templates suggested by the clustering algorithm for the white water valve, in decreasing order of their sizes.	59
3.3	ROC curves of the white water valve for three templates. The operating point is at the right bottom corner with a 1.0 defect detection rate and a 0.0 false alarm rate.	60
3.4	The template chosen for the red water valve.	61
3.5	OK templates suggested by the clustering algorithm for the red water valve in decreasing order of their sizes.	61
3.6	ROC curves of the red valve for one and four templates. The operating point is at the right bottom corner of the curve drawn with four templates, with a defect detection rate of 0.9 and a false alarm rate of 0.2.	62
3.7	The handles of the clamps of the hose vary in their appearance and sometimes, are not visible at all.	63
3.8	The different template options we had.	64
3.9	The template chosen for the tub hose with the clamp ring and the hose visible.	64

3.10	OK templates suggested by the clustering algorithm in decreasing order of their cluster sizes.	65
3.11	ROC curves of the hose for three templates and six templates. The operating point is at the right bottom corner for the curve drawn with six templates, with a defect detection rate of 1.0 and a false alarm rate of 0.05.	65
3.12	Different options we had for choosing the template for the EMI filter.	66
3.13	An OK and an NOK image of the EMI filter.	66
3.14	The template chosen for the EMI filter.	66
3.15	OK seeds of the EMI filter suggested by the clustering algorithm in decreasing order of their cluster sizes.	67
3.16	NOK seeds of the EMI filter suggested by the clustering algorithm in decreasing order of their cluster sizes.	67
3.17	EMI filter as a 1 class problem using 4 OK training templates detects around 80 % defects with a false alarm rate of 3.0.	68
3.18	EMI filter as a 1 class problem using 7 NOK training templates detects around 70 % defects with a false alarm rate of 3.0.	69
3.19	EMI filter as a 2 class problem combining the results from 4 OK and 7 NOK templates has a peak defect detection rate of 0.97 with 2.9 false alarms per defect.	70
3.20	The template chosen for the pressure sensor.	71
3.21	ROC for the pressure sensor. With just 1 template the system was able to detect all the defects with 0 false alarms.	72
3.22	The template chosen for the wire holder.	73
3.23	ROC for the wire holder using 34 training templates. The system was able to find only around 50 % of the defects at a high false alarm rate of 5.0.	74
3.24	The thermistor screws appear larger if closer to the camera.	74
3.25	The template chosen for the thermistor screws.	75
3.26	OK seeds of the thermistor screws suggested by the clustering algorithm in decreasing order of the sizes of their clusters.	75
3.27	ROC for the thermistor screws using 4 training templates. At a 2.0 alarms per defect, the system was able to achieve a 0.83 defect detection rate.	76
3.28	The template chosen for the screw near the thermistor.	77
3.29	OK seeds of the thermistor screws suggested by the clustering algorithm in decreasing order of their cluster sizes.	77
3.30	ROC for the screw near thermistor using 3 training templates. At the operating point, the defect detection rate is 1.0 and the false alarm rate is 0.56.	78
3.31	The template chosen for the screw on the back.	79

3.32	ROC of the top left screw at the back of the machine for 31 templates. The system achieves a 1.0 defect detection rate at 1.02 false alarms per defect detected.	79
4.1	Appearance of defects for the wire holder use case. In the first figure, the wire harness is inside the wire holder (OK). Next, the wire holder is not tied and the wire is left dangling outside of the holder (NOK). Next, the holder is not tied and the wire can slip out (NOK). In the fourth figure, the wire is not going through the holder and in the last figure the holder is absent (NOKs).	82
4.2	The training templates suggested by the clustering algorithm for the wire holder. The only defect captured by the algorithm is image number 33 where the holder is untied and the wire is dangling.	83
4.3	Decision tree to help the human engineer train and evaluate the inspection system for appearance variability.	85
4.4	Decision tree to help the human engineer tweak the inspection settings.	86

Chapter 1

Introduction

This work is in collaboration with Samsung Electronics Home Appliances (SEHA) and it has been going on for the last 1.5 years. It considers the problem of variability in the appearance of machine parts, while performing automated inspections, in an appliance manufacturing plant; in this case, a washing machine manufacturing plant.

A manufacturing plant is an industrial site where workers or robots manufacture goods, or process one product into another [9]. In an appliance manufacturing plant, the end-product is an appliance or a piece of equipment designed to perform a specific task. The manufacturing process is called an assembly line, where machine parts are fitted together and the unfinished assembly moves from one work station to another on a conveyor belt, until the final product is built. In the context of this thesis, machine parts are metal or plastic pieces used to assemble washing machines. They can be valves, connectors, clamps, holders and fasteners. Parts are inspected to find any defects which might have occurred while assembling them. Learning the variability of the appearance of different parts to be inspected is very important for finding defects with high accuracy and precision.

If overlooked, defects might be difficult to find as you go further down the assembly line. A defective part or a defective assembly is bad for the manufacturing company. A defective machine has to be repaired or replaced. Repairing requires skilled labor, which is expensive and replacement is a big loss to the company. It also affects consumer trust and the brand value of the company. Hence, any manufacturing company needs to have an accurate and precise inspection system in which parts are inspected at each work station and repaired as soon as a defect is found, and defects are not allowed to propagate down the assembly line.

1.1 Camera-based inspection system

An inspection is a careful examination or scrutiny of something to make sure that rules are followed and standards are met. Traditionally, inspections at manufacturing plants have been performed by people called inspectors. There can be an inspector at each workstation on an assembly line who inspects the part assembled by another human or a robot for defects. If they find a defect, the conveyor belt which moves the assembly is stopped for repair. There are two main disadvantages to the approach: the cost of hiring inspectors is high and the performance of humans decreases due to vigilance decrement.

Recently, camera-based inspection systems have been developed to overcome the shortcomings of the traditional approach. In this system, cameras are mounted on the assembly line to inspect the parts assembled at each work station. There can be a single or multiple cameras capturing images of the part from different angles after it has been assembled. The camera can be a smart camera with a processor of its own or just a normal camera that is connected to a computer via Firewire, USB or Gigabit Ethernet. There can also be a PLC based sensor system to detect when

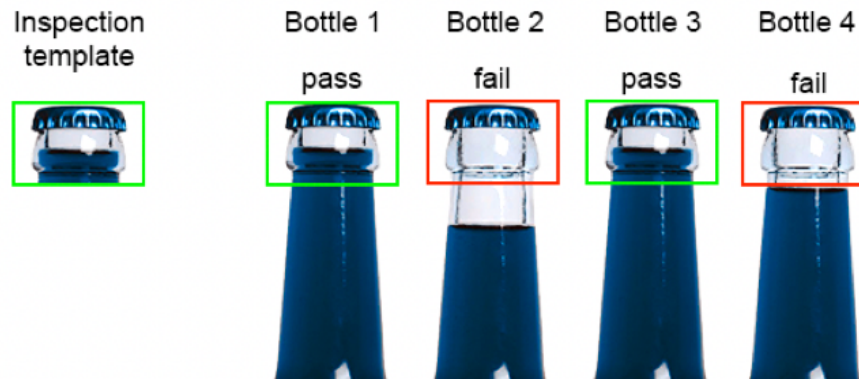


Figure 1.1: A fill-level inspection. In Bottles 2 and 4, the fill-level of the beverage inside is below a threshold and hence they fail the inspection, while bottle 1 and 3 are full and they pass.

the assembly is in frame and to trigger the camera to start inspecting different parts.

The camera-based system needs to be provided with an image of the part without any inherent defects, and a proper assembly. This image is also known as the inspection template. The live image of the part on the assembly line or the image of the part to be inspected, is then compared with the inspection template to check for similarities. If the live image matches the inspection template well then it passes, else it fails and the assembly line is stopped until the part is repaired or replaced. This process is shown in figure 1.1 for a fill-level inspection system.

A fill-level inspection system has to check if the beverage in a bottle is filled up to a certain level. The first image on the left is the inspection template and the four bottles on the right are live inspections. It can be seen that bottles 1 and 3 match the inspection template well and pass, whereas, bottles 2 and 4 in which the beverage is filled less, fail.

The fill-inspection system discussed above requires just one inspection template to perform live inspections. Sometimes the part to be inspected might have some variability in its appearance and one inspection template is not enough to capture

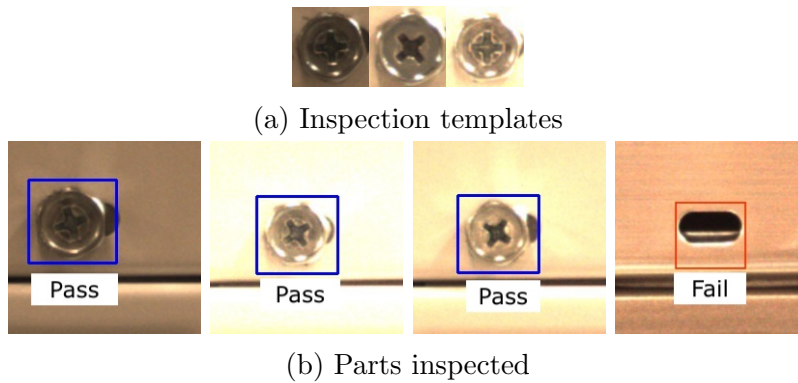


Figure 1.2: Screws inspected in the first three images match the three inspection templates well and pass the inspection. The fourth image has a missing screw and hence the inspection fails.

this variability. The part could vary in its color or orientation and the system needs to be fed with multiple inspection templates that capture these different variations in the appearance of the part.

Figure 1.2 shows the inspection templates and live inspections for a screw inspection system. There is an inspection template provided to the system for each of the different variations in the appearances of the screw and the background frame. If one or more templates match with the part really well then the system assumes there is no defect or passes, else it signals a defect or fails.

1.2 Part appearance variability

As mentioned earlier, a camera based inspection system needs to be provided as many inspection templates as the number of distinct appearances of the parts to be inspected.

Sometimes, one of the variations in the appearance of the part to be inspected might not get considered. This might cause the system to fail, give false alarms and

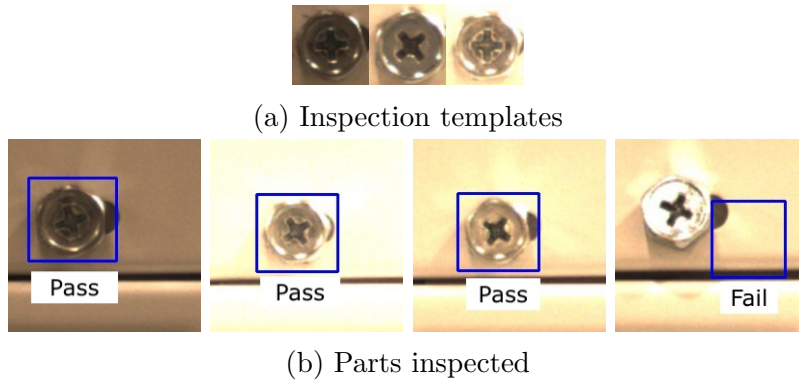


Figure 1.3: The screw in the fourth image fails the inspection because there is no inspection template that matches it well. The system erroneously stops the assembly line.

erroneously stop the assembly line. As shown in figure 1.3 the system is provided with three different templates for three possible appearances. Even though the screw in the fourth live image is properly fastened, it does not match well with any of the inspection templates provided to the system, causing the system to fail.

Thus the system fails due to a particular variability in the appearance of the screw not known to it, and this is called the problem of part appearance variability. Any camera based inspection system has to be trained for this variability in part appearance to avoid such false alarms, and increase the defect detection rate.

1.3 Training for part appearance variability

To train a camera based inspection system for part appearance variability, typically, a human engineer at the factory would try to anticipate expected variations based on the different appearances of the parts. These images with different appearances would then be fed to the system as inspection templates. This section explains why this training process is difficult, costly and time consuming.

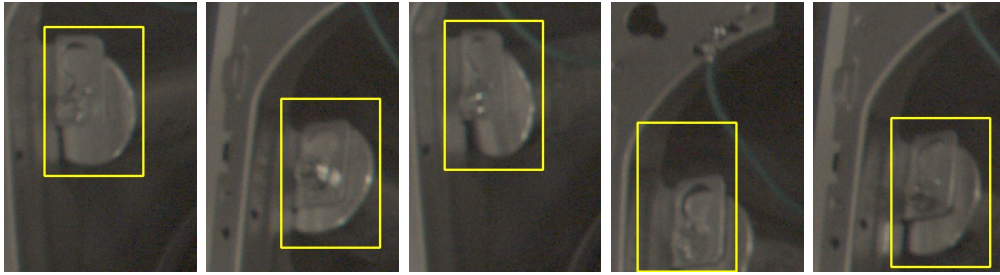


Figure 1.4: Variability in appearance due to change in viewing angle.

Machine parts could have a lot of variability in their appearances due to: changes in viewing angle, parts incorrectly assembled, changes in color, occlusions or changes in orientation of the parts. Each of these are explained in detail below, with an example.

Figure 1.4 shows an example where the variability in appearance is due to a change in the viewing angle of the camera. The machine might not always be in the same position when a frame is captured and the part to be inspected could be closer or further away from the camera. This change in the viewing angle leads to changes in how the part looks. It can be seen in figure 1.4 that there is variability in the appearance of the part in the live images and none of the appearances are defects.

Sometimes there is an error in the assembling that causes the part to look different. In figure 1.5 the second and third image look different because they haven't been assembled properly. In both the images, the part has not been fastened down properly and looks loose. Thus the variability in appearance in this case is due to incorrect assembling and could be easily overlooked by a human.

Parts and the frames on which they are fastened or attached can have different colors and specularities. Figure 1.6 shows eight different appearances of the screws. Each image has variation in the color of the frame, color of the screw and in the appearance of specularities caused due to the lighting in the factory. These colors and specularities make the parts look different and could affect the performance of



Figure 1.5: Variability in appearance due to incorrect assembling.

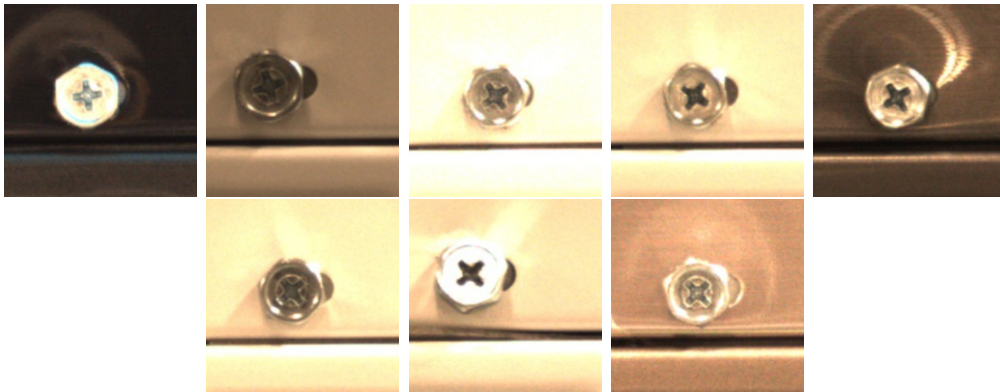


Figure 1.6: Variability in appearance due to change in color of background, the color of the part and specularities.

the system. It is difficult for a human to identify these different appearances and anticipate their effect on the performance of the system.

Another reason for variability in appearance could be occlusions. When something obstructs or hides the view of a part, as seen from the camera, it is said that the part has been occluded or hidden or obstructed. This can cause variability in appearance as shown in figure 1.7. There is a green wire occluding the part in all the images. In one of the images the color of the wire is black and the orientation of the wire appears different in each of the seven images. The part is tilted to the right or left and is not exactly upright in most of the images. This also leads to change in appearance and it is difficult for a human to grasp these subtle changes.

Thus we can see that there a lot of variabilities in the appearance of different

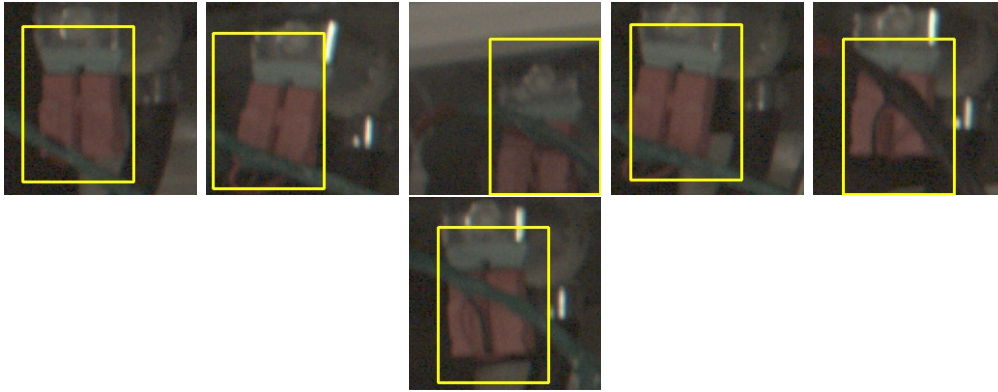


Figure 1.7: Variability in appearance due to change in orientation and occlusions.

parts and sometimes they occur in combinations, which makes the problem even harder, for a human eye. In order to train the system for these complex variabilities in appearance, the human operator at the factory would have to spend a lot of time looking at tens of thousands of images of a part and find inspection templates that capture the variability in its appearance. They would have to do this for each and every part that needs to be inspected.

This is a time consuming and an error prone method and there might be a particular appearance or different appearances of a part that did not catch the eye of the human, which might cause the system to fail, as seen in figure 1.3. Some parts might have a lot of variations in their appearances and it would be really difficult for a the human eye to catch all the different cases. The system might trigger a lot of false alarms if there are not enough inspection templates that capture these variabilities in part appearance. This would stop the assembly line more frequently and reduce the throughput of the factory; thereby, making the inspection system very unreliable. Moreover, hiring human engineers for this process is a costly affair for the manufacturing company.

This thesis explains a novel approach of training a camera based inspection system to solve the problem of part appearance variability.

1.4 Background

This section gives a detailed overview of the work done by our group in the last 1.5 years to solve the problem of appearance variability or to train an inspection system for appearance variability [7]. It explains the core technologies used, the methods developed, and the inspection system prototype that was built to test these methods.

1.4.1 Template matching

Camera based inspection systems use an image processing technique called template matching for comparing the inspection templates to the live images. Template matching is a technique of finding a small part in a large image that matches a small template image. In the context of part inspections, the template image is the image of the part to be inspected, in a perfectly good condition, and, the large image is the image of a portion of the assembly where it is expected that the part would be present. In this thesis, we call the template image as the inspection template (IT) and the large image as an inspection search window or just an inspection window (IW).

As discussed earlier, in order to inspect a part, the part inspection window is compared against all its inspection templates and depending on how well the window matches the templates, it passes and is labelled as an OK (good), or it fails and is labelled as an NOK (defect). The inspection template is slid/convolved over the entire inspection window and compared, to find out the location of the inspection template or something that looks like the inspection template, in the inspection window. A match score is generated for each point inside the inspection window. It is in the range of 0.0 to 1.0, where 0.0 indicates a total mismatch and 1.0 indicates a perfect

match. The point with the maximum score is regarded as the match location of the template inside the window. This sliding/convolving of the inspection template over the window is called template matching.

In this thesis, template matching is implemented using OpenCV, an open source computer vision and machine learning library [3]. The library provides various comparison or score generating methods for template matching. For the purpose of this thesis the normalized cross correlation (NCC) provided by the `TM_CCOORR_NORMED` function is used for template matching. The algorithm produces two outputs: the best match location and the corresponding match score.

The template matching function in OpenCV is modified to generate a third output; a gray scale image called the template match image. It helps us in deciding if a particular template is appropriate for an inspection problem. This image is generated by scaling the match scores at each and every point in the inspection window to a scale from 0 to 255 where 0 is black and 255 is white and everything in between is some shade of grey. This scale is called the grey scale and the image generated using this scale is called a grey scale image. The bright spot in the image is the location with the highest score, in other words, the best match location. Ideally, there should be a single bright spot and everything else should be black or a dark shade of grey. Too many bright spots or patches indicate a poor choice of the inspection template. It indicates that too many points in the original image had a score close to the highest match score, in other words, it shows that the template is not distinct enough and is matching multiple locations in the image. Such a template would be bad for an inspection system which needs high precision and accuracy.

Figure 1.8 displays the three outputs generated by the algorithm for a particular use case. In this example the template matching algorithm is used to inspect a part called hose clamp. Thus the inspection template in this case is an image of the

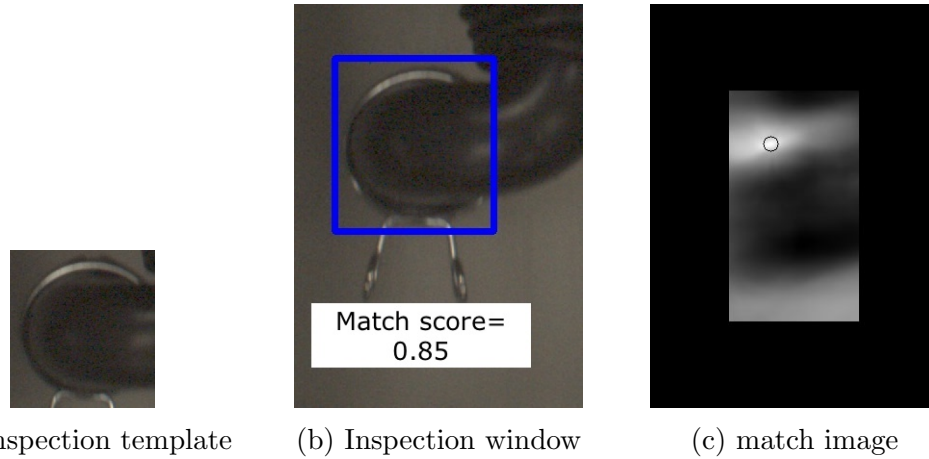


Figure 1.8: Template matching using RGB images. The hose template matches with a score of 0.85 at the best match location (center of the part) as seen in the inspection window. A bright white spot can also be seen in the template match image on the right at the same location.

hose clamp in a perfectly good condition. The IT is slid/convolved over the inspection window (region where it is expected that the part is present) and a match score is generated for each and every point in the window. A blue box with dimensions equal to that of the template is drawn with its center as, the point with the highest match score, or, in other words, the best match location of the template inside the inspection window. In this case, the score at the best match location is 0.85 and a bright white spot can be seen in template match image at the same location as seen in figure 1.8c.

As mentioned earlier, the template matching algorithm has three different outputs: the match location, the match score and the match image. Sometimes, to make the process color and light independent, gradient or edge images are used instead of the original RGB images to generate these outputs [10]. Depending on the type of inspection, either or all the outputs could be generated using the original images or the gradient images of the IT and the IW like in figure 1.9.

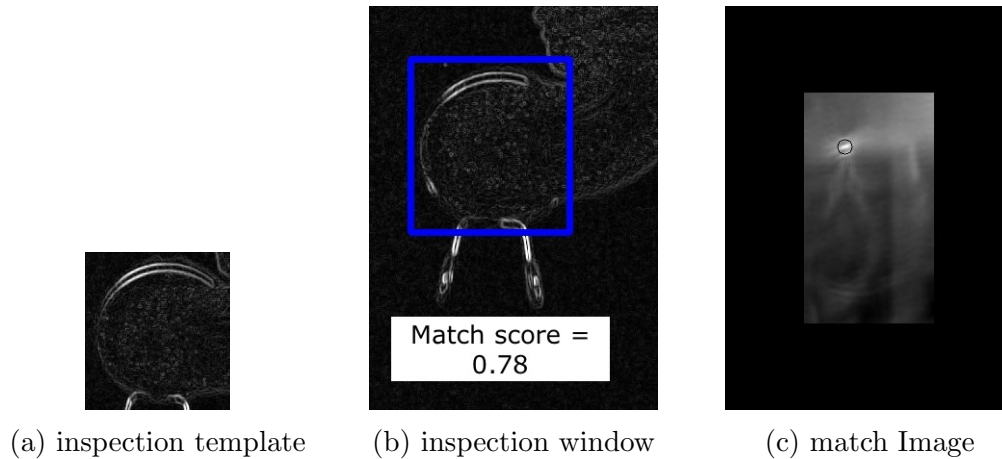


Figure 1.9: Template matching using edge images. The hose template matches with a score of 0.78 at the best match location (center of the part) as seen in the inspection window. A bright white spot can also be seen in the template match image on the right at the same location. The white spot in this case, is more distinct.

1.4.2 The inspection system prototype

In order to test of our methods, our group built a prototype of an inspection system. The system is capable of inspecting parts in motion and raising an alarm if a defect is detected. It can also be used to collect data (images) into a windows laptop during normal assembly operations.

1.4.2.1 Hardware

The system consists of the Basler acA2500-14gc [2], a windows laptop and LAN and power chords required for Power over Ethernet (PoE) [11]. The Basler camera has resolution of 2590x1942 pixels and captures frames at the rate of 14 frames per second. For PoE, power from a source and data from the windows laptop is transported over a CAT 6 cable via a PoE injector.

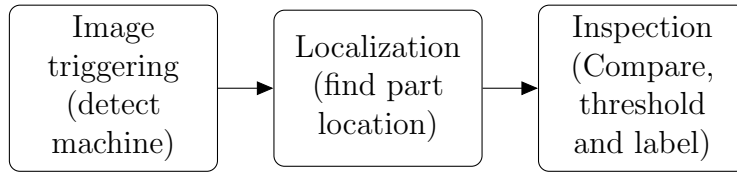


Figure 1.10: Overview of the inspection process.

1.4.2.2 The inspection process

The inspection process involves three stages and all three use template matching as their core algorithm. The first stage is detecting if the machine is in the view of the camera. This is done by something called as image triggering. The second stage involves finding the location of the inspection windows of the parts to be inspected, also called as localization. The third stage is where the parts are inspected and the assembly line is stopped if a defect is found. Figure 1.10 gives an overview of the entire process. Each of these stages are discussed in detail in this section.

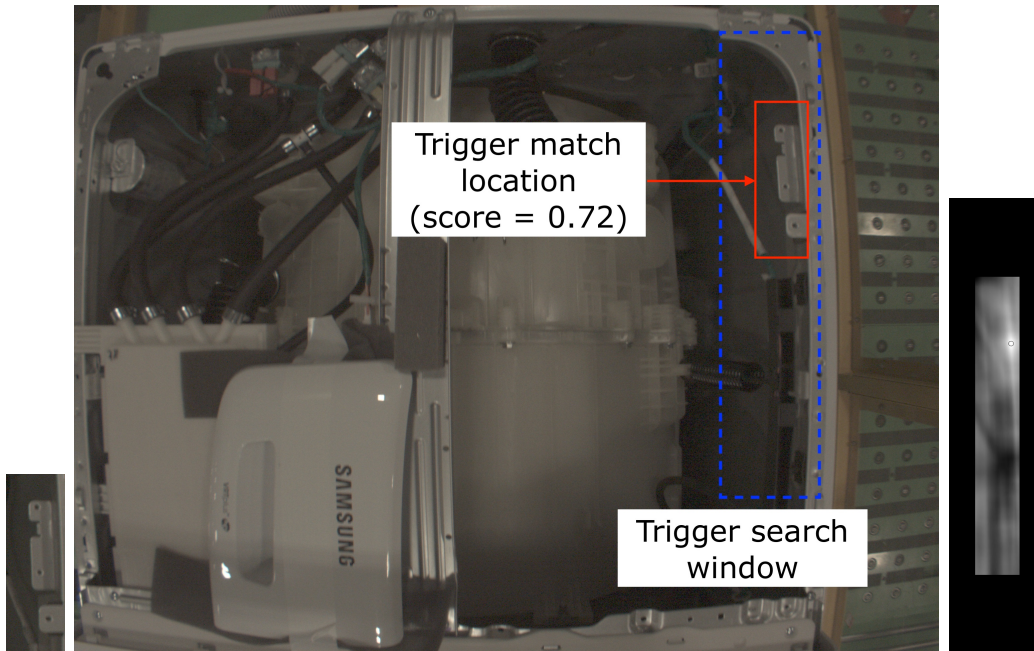
Image triggering This is the first stage in the inspection process. The unfinished assembly or the machine in an appliance manufacturing plant stops at each work station where different parts get assembled by a human or a robot. Each time it passes a work station the assembled part or parts need to be inspected. Thus the inspection system needs to be installed immediately after a work station and it needs to be notified when the machine is in view. This is done by using image triggering.

An image trigger in an image, in the context of this thesis, is that which initiates the inspection process. Just like the inspection template and the inspection window discussed earlier, we have a trigger template and a trigger search window. A trigger template is chosen such that it is present on each and every assembly on that particular assembly line. We typically select part of the background housing as shown in the red box in figure 1.11a. The trigger template is expected to be found

inside the trigger search window. The size of the search window is chosen after some trial and error; after understanding the variability in the orientation of the assembly when it is in the frame of the camera. The size of the trigger search window is set in such a way that it is not too large. A large trigger search window might slow down the image triggering process which might lead to missed inspections if the conveyor belt moves too fast.

This can be better explained using an example. Figure 1.11a shows the trigger template and a frame captured by the camera. The small image on the left is the trigger template, the blue dotted box in the image in the middle is the trigger search window, and the red box is the location of trigger where it was found with a match score of 0.72. The location of trigger in the window is indicated by the bright spot (this is the center of the red box) in the template match image, which is the rightmost image in figure 1.11a. Figure 1.11b shows where the system was setup at the factory for this use case. Basically, the machines keep moving from left to right and the live inspection system keeps searching for the trigger, in the trigger search window.

Part localization After detecting that the machine is in frame, the next step in the process of inspection is to locate the parts that need to be inspected. Assuming that all machines are assembled in a similar fashion, all parts are at fixed locations in the machine. Thus their distances from the image trigger are the same. The distances along the x axis are denoted as dx and those along the y axis are denoted as dy . Each part has a dx and dy from the trigger that is input to the system before starting the inspection process. Thus if the the location of the top left corner of the trigger inside the trigger search window is at x,y , then the top left corner of the inspection window for a part can be defined at $x+dx,y+dy$. The sizes of the inspection windows are pre-defined. An approach similar to the trigger search window is used for deciding the size



(a) The trigger template, the frame of the machine captured by the camera and the trigger match image. The red box shows the location of the trigger found inside the frame and the blue dotted box shows the trigger search window. A bright spot can be seen at the match location (center of the red box) in the match image.



(b) Camera installation at Samsung Electronics Home Appliances, Newberry, SC.

Figure 1.11: Camera installation and a frame captured from the camera.

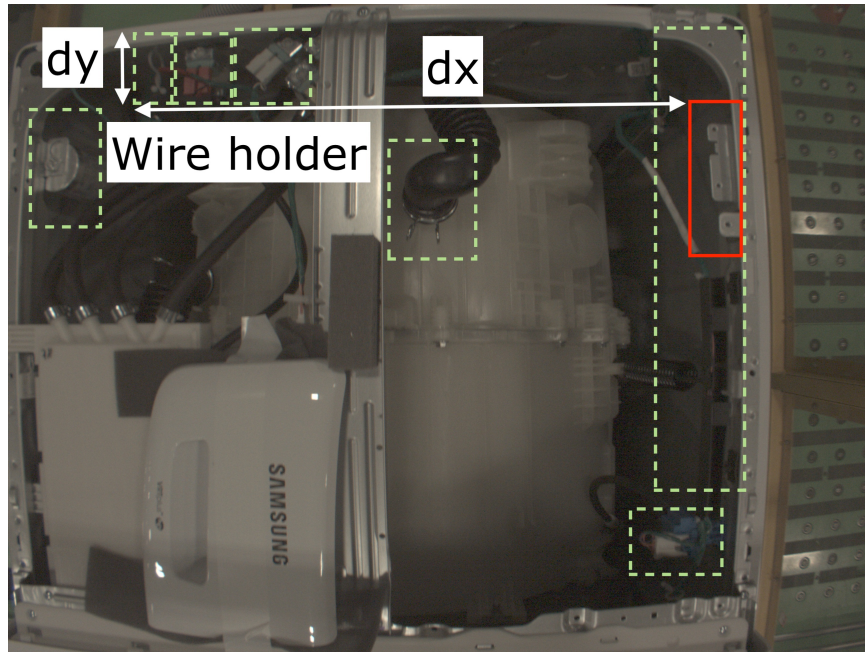


Figure 1.12: After locating the trigger template shown in a red box, the system locates different parts shown in green dotted boxes. The top left corner of the windows are at a fixed dx and dy from the trigger template. This is shown for the wire holder in this case.

of the inspection window. In this case, we look at the size of the inspection problem (part) and its orientation in different frames. Different parts can have inspection windows of different sizes as seen in figure 1.12. Figure 1.12 shows the machine with different parts. The blue dotted boxes are the inspection windows of different parts that need to be inspected for defects. The dx and dy for the part called wire holder is shown in the figure. Both, dx and dy have negative values in this case because the part is on the left and above the image trigger.

Part Inspection Once the machine is frame and the system knows the probable locations of different parts, it is time to inspect. Inspection templates for each part are fed to the system before starting the inspection process. Each part can have varying number of inspection templates. The inspection templates for a part are compared

with the respective inspection window using template matching and a template match score and a match image is generated for each template. If the template that matches the best, (has the highest match score and the brightest spot in its match image) has a match score lower than a threshold (for that part) then it is assumed that there is a defect and the part is labelled as an NOK and an alarm is raised, else it is an OK.

As shown in figure 1.13, after locating the inspection window of the white water valve using part localization, the templates of the white water valve are compared with the window using template matching. The template with the highest score of 0.85 is shown on the left and its match image with the best match location indicated by a bright spot is shown on the right of the machine. The match location is shown in the machine with a red box with size equal to the size of the template. The score of 0.85 is greater than a threshold and hence it is assumed that the part is OK or has no defects. This method is implemented for all the parts shown in figure 1.12 simultaneously as soon as the machine is in frame.

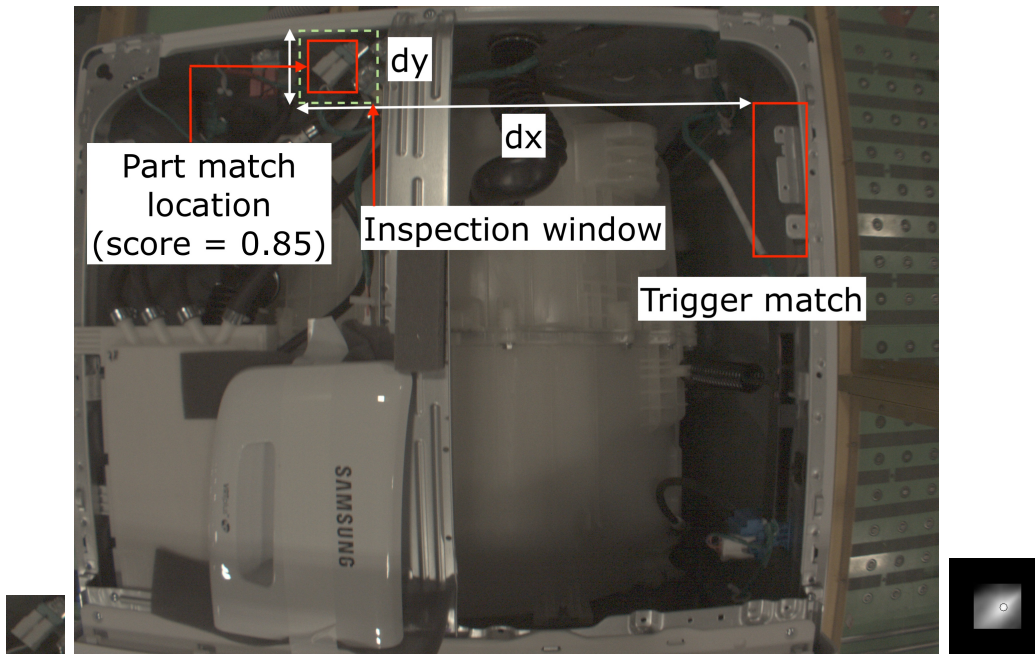


Figure 1.13: A frame of the machine with different parts. After locating the image trigger, the system has defined an inspection window for the white water valve at a fixed distance from the image trigger. A match score is generated for the part (white water valve) inside the inspection window. The match score (0.85) is higher than a threshold and is hence labelled as an OK. The template match image of the inspection template and the window is also shown.

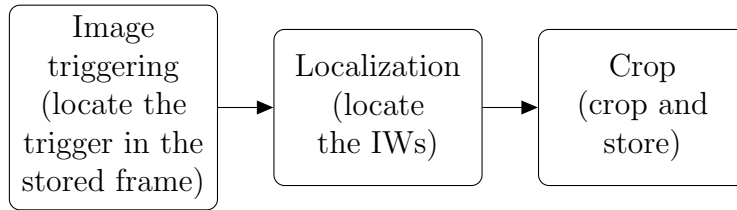


Figure 1.14: Overview of the cropping process.

1.4.3 The data collection process

Data is collected to test, and further develop our methods of training the inspection system for appearance variability. The data collection process developed by our group is similar to the inspection process and is performed during normal assembly operations. Once the machine is in frame, as detected by the image trigger, instead of localizing and inspecting different parts, the frame is stored in a database (windows laptop).

1.4.4 The cropping process

After the data has been collected, we need images of the inspection windows of the different parts to be inspected. For this reason, a cropping program which crops out inspection windows from all the frames (data) collected in the data collection process, was developed by our group.

This process is again, similar to the inspection process. As the inspection windows of parts are defined with respect to the location of the trigger in the image, image triggering is performed in order to find the location of the trigger. Next the location of the top left corner of the inspection windows of different parts are found out and the parts are cropped. Figure 1.14 gives an overview of this process.

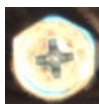
1.4.5 Clustering to find appearance variability

As discussed earlier, it is a time consuming task and an error prone task for a human to go through each and every inspection window and find out the different variations in the appearance of each and every inspection problem. In order to solve this problem a method to automatically learn part appearance variability was developed by our group at Clemson [7].

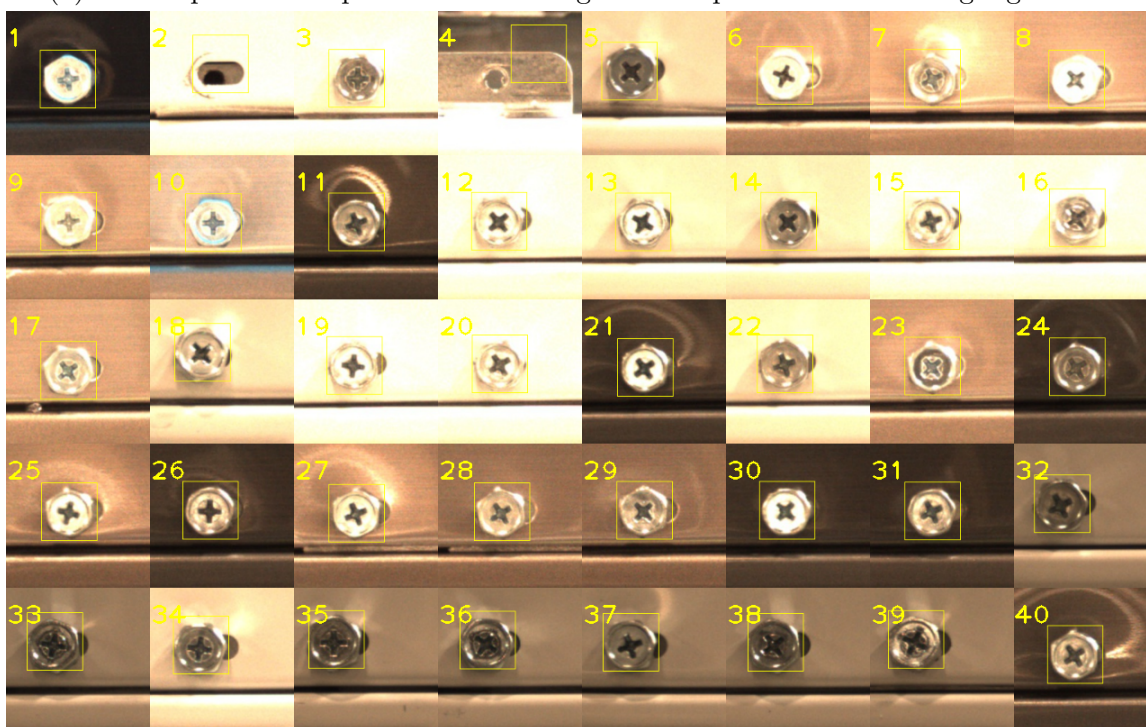
It is based on an unsupervised machine learning algorithm known as clustering and automatically learns the variability in appearance of the parts to be inspected. It takes in as input, all the inspection windows generated by the cropping program, an inspection template of the part to be inspected and a configuration file. It then forms different clusters or groups of IWs that have similar part appearances. Each cluster has a seed which represents the variability in the appearance captured by that particular cluster; in other words, all parts in that cluster look more or less the same in comparison to the seed.

Now, instead of looking at each and every image, the human engineer can just look at the seeds and pick the ones that vary in appearance and are not defects. The method is generic and not problem specific and automatically suggests inspection templates that capture the variability in appearance of parts to be inspected; it does not require any human interference once given the required inputs.

Figure 1.15 shows the output of the clustering algorithm for the screw use case. It takes in as input the inspection template as shown in figure 1.15a, and approximately 3,000 inspection windows of the screw. Different variations in the appearances of the screw can be seen in figure 1.15 as found by the algorithm. Inside each of the windows there is a bounding box that highlights part. The size of these highlights is same as the size of the input template, as shown in figure 1.15a. These



(a) The inspection template of the screw given as input to the clustering algorithm.



(b) A collage of the 40 cluster seeds of the screw generated by the clustering algorithm.

Figure 1.15: The 40 cluster seeds capture different variations in the appearance of the screw including defects.

are called seed templates or seeds. These seed templates of all the clusters can be suggested to a human engineer as probable inspection templates which could be used for live inspections as seen in figure.

1.4.5.1 The clustering algorithm

This section reviews the clustering method described in [7]. The system is based on a variant of the k-means clustering algorithm. The k-means clustering algorithm divides a large dataset into k groups because there are k seeds or k points within the data set that are manually chosen with some heuristics. The algorithm has two steps: assignment and update. In the assignment step, each data point is assigned to one of k seeds. Each data point is assigned to the seed which is the closest to it, in terms of a distance metric. At the end of this step we have k groups with their respective k seeds, where the seed of a group is one of the points in the group that represents the group. In the update step these k seeds are updated so as to better represent their group. Next, the algorithm goes back to the assignment step and this loop of assignment to update to assignment goes on until a stopping condition is met. At the end, k distinct groups of points are formed, where each group has a distinct feature. In the case of clustering inspection windows of a part, the k seeds would represent the k distinct appearances of the part.

The algorithm developed by our group takes in as input all the inspection windows of a part, an inspection template and a threshold that defines the stopping condition. Unlike k-means clustering where the user provides k different seeds, the algorithm has to be given only one seed which is the template (one ideal appearance of the part) of the part to be inspected.

At the beginning, in the assignment step, all images are assigned to one cluster. A match score and a match location is generated for all the inspection windows

after comparing them with the seed using template matching, and the variance of all the match scores is calculated. This is the first cluster or group of images formed by the algorithm. In the update step, the least scoring member of this cluster (part which looks very dissimilar to the seed) is assigned as the second seed. Now, the scores and locations are calculated with respect to both the seeds for all the inspection windows by using template matching. Next, in the assignment step, cluster membership is assigned to a window according to the seed which has a better match score for that particular window. At the end of this step, we have two clusters with their two respective seeds. Now, the variances of scores for cluster one and for cluster two are calculated. In the update step, the least scoring member of the cluster with the maximum score variance (cluster which has a lot of variability in the appearance of its members) is assigned as the the third seed. Now, the scores and locations are calculated with respect to three seeds for all the inspection windows. This loop goes on until the cluster with the maximum variance is below a certain threshold called the maximum cluster variability threshold or MCVT. At the end, the algorithm generates k different groups of inspection windows with their match locations and match scores with respect to their respective k seeds. Each seed represents the variability in appearance of the part captured by its group or cluster. Information about each cluster such as the seed, number of members, score variance, match locations and scores of each member are stored in a text file.

Unlike k-means clustering the value of k is not pre-defined. It depends on the MCVT, which in turn, depends on the complexity of the inspection problem and is calculated using normalized difference accumulated image (NDAI) of the part to be inspected. The entire algorithm, along with the tuning of MCVT using the NDAI, is discussed in more detail in [7].

1.5 Related Work

W. A. Perkins developed a system that could classify different parts and detect defects at the same time. There were two parts to the inspection system, one was determining inspection regions of a part and then determining test parameters for those regions in order to inspect the part [5]. C. Tsatsoulis and King-sun Fu developed a programming language named SAIL (simple assembly inspection language) for visual inspection of assemblies. The system is based on measurement of different characteristics of parts. It is described in [8]. Progress in the field of machine vision and an overview of the real-world applications can be found in [4]. It describes popular software and hardware tools required to build a machine vision system. Different image registration methods are discussed in [13]. The most recent work which is closely related to what we have done is a patent by Cognex Corporation [6].

All the above works are part specific, whereas, we are trying to solve the problem using a generic approach which can be applied to any inspection problem. We also consider the problem of a human engineer at a manufacturing plant being able to use our methods.

1.6 Novelty

The novelty of this thesis lies in three components of the inspection system:

- **Training the inspection system for appearance variability:** Previous work developed a tool (clustering) that could learn part appearance variability [7]. We performed a set of experiments, trying different ways of using the tool to train an inspection system. Variations included how long to run the clustering, how to select inspection templates from its output, whether or not

to use NOK templates in addition to OK templates, and when and how to tweak the inspection settings (pre-processing, IW size, IT size). The goal was to maximize the defect detection rate while minimizing the false alarms. Lessons learned from performing these tests on 9 different inspection problems were used to build a decision tree that could be used by engineers for training the inspection system on future inspection problems. We also report our inspection accuracies and the training process for all the 9 inspection problems.

- **Data collection:** Previous work collected 1,052 images for 3 inspection problems. In this work we collected a large amount of new data: 39,268 frames for 6 inspection problems (EMI filter, red water valve, white water valve, hose clamp, wire holder and the pressure sensor), 23,247 frames for 2 more inspection problems (thermistor, screw near thermistor) and 3,987 frames for another inspection problem (back screw). A total of 249,371 images were collected for 9 inspection problems.
- **The clustering algorithm:** The clustering algorithm developed previously [7] was modified to allow it to run on arbitrary sized data sets. This was needed because the data we collected was orders of magnitude larger than in previous work.

Chapter 2

Methods

Figure 2.1 overviews the methods used for our experiments in order to train an inspection system prototype for part appearance variability. After collecting data, parts to be inspected are cropped and manually reviewed for defects (ground truth). The clustering algorithm is run on the crops of the parts to be inspected. The clustering algorithm outputs all the cluster seeds or the different appearances of the parts. The cluster seeds are then used as training templates to train and evaluate an inspection system prototype by comparing its results with the manually reviewed data. Different pre-processing and training options are tested to improve the results our methods with a goal of reducing false alarms and increasing the number of defects detected. Each of these processes are discussed in detail in this chapter.

2.1 Data collection

We collected data on three different occasions. The first data collection occurred over a period of 3 months as we tested different variations of pre-processing and camera settings. The second data collection occurred at another assembly line

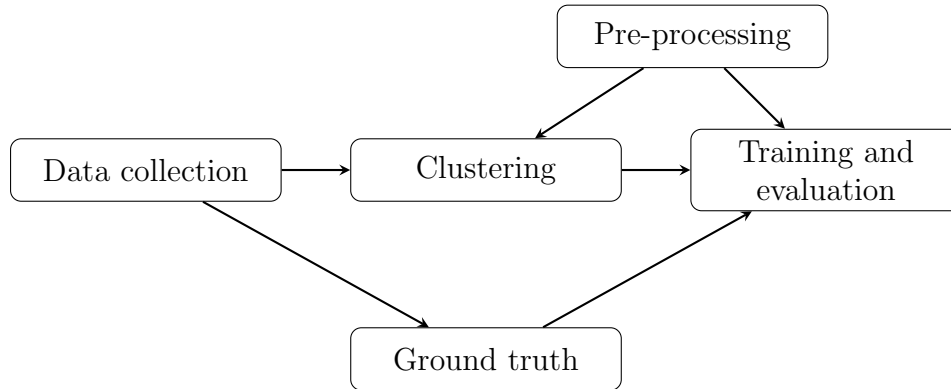


Figure 2.1: Overview of the methods used for our experiments.

location for 2 new parts. The third data collection used cameras already installed at SEHA and being used by their existing Q-eye inspection system. All the data collected has been summarized in table 2.1.

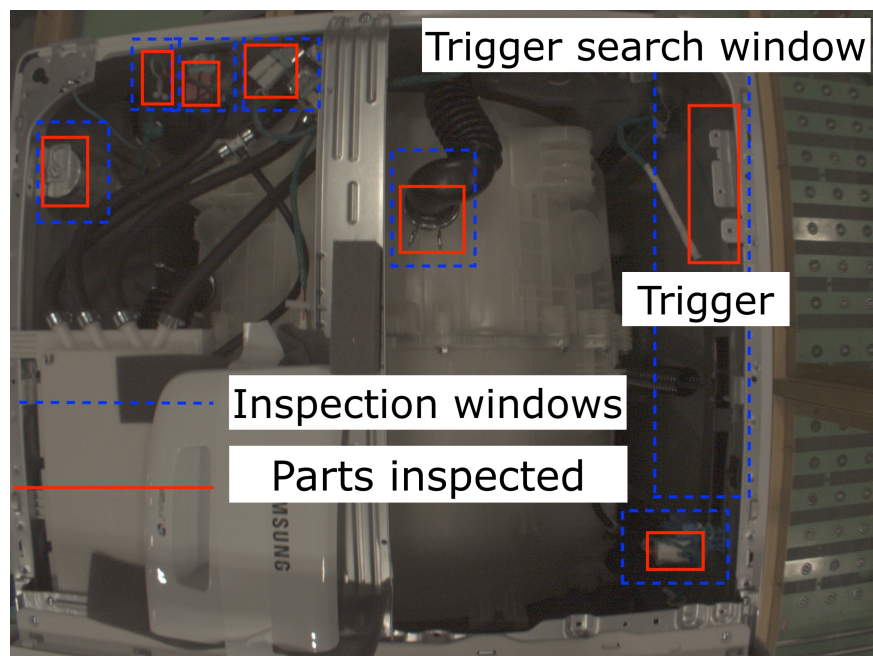
2.1.1 The low gain data set

This data set contains parts from the frame shown in figure 2.2. A total of 39,268 frames were collected, 2 frames per machine. Out of the 12 parts in the frame that needed inspection, we chose only 6 to do our experiments on due to occlusions or poor lighting conditions. This section describes each of the 6 parts and what they were inspected for, viz., the white water valve, the red water valve, the pressure sensor, the tub clamp hose, the EMI filter and the wire holder. Each of these can be seen in figure 2.2.

White water valve Valves in a washing machine are mechanical devices that control the amount of flow of hot and cold water inside the washer. They are connected to water hoses that transports the water inside the washer. The valves have ports which

Data-set	No. of Machines	Start date	End date	Notes
Pilot	3,347	12/6/2018	12/10/2018	First collection test.
Low Gain	19,634	1/11/2019	1/29/2019	Largest dataset. After collection, it was determined that the camera gain was too low (dark images). (figure 2.2)
High Gain	3,378	2/11/2019	2/12/2019	Adjusted gain and exposure time to increase image brightness.
New Model	4,520	2/12/2019	2/21/2019	New washing machine model. Production slow (about 1 machine/10 minutes).
Thermistor	11,625	9/26/2019	10/10/2019	Most recent data collection to demo our system to the Samsung executives. (figure 2.9)
SEHA	3,987	03/09/2019	03/11/2019	This is the data provided by SEHA (figure 2.12)

Table 2.1: Data collection summary.



(a) Frame of the machine with inspection and trigger search windows predefined.



(b) Pressure Sensor



(c) EMI Filter



(d) Red Water Valve



(e) Wire Holder



(f) Tub Clamp Hose



(g) White Water Valve

Figure 2.2: Low Gain dataset: Six different parts inspected in 39268 such frames.



Figure 2.3: In the first figure, the white valve is present and fully secured and connected to the green mount (OK). Next, it is absent (NOK). In the third figure, it is dangling (NOK).

are used to send electric power to the valve to open and close the flow of water based on the user defined settings of temperature and volume. These ports have solenoid valves which carry electric current. Incorrect assembling of the solenoid valves might lead to the tub not getting filled properly or the temperature of the water not being as defined by the user.

The white valve is inspected to make sure it is present and connected to the green mount as shown in figure 2.3. In this thesis we focus on detecting presence or absence of the valve. We do not try to find whether the valve is partially or fully connected, we are just inspecting to make sure it is connected and not dangling or totally absent.

Red water Valve The red water valve is similar to the white water valve, but red in color. It is also inspected to make sure it is present and fully secured and connected to the green mount and not dangling or absent, as shown in figure 2.4.

Tub Clamp Hose A hose is a tube that transports liquids from one location to other. In this case, it is the hose that provides water to the tub of the machine. The hose clamps are used to attach the hose to the tub via a fitting such as a barb. The hose is inspected to check whether it is fully seated on the tub and that the clamp is



Figure 2.4: In the first figure, the red valve is present and fully secured and connected to the green mount (OK). Next it is absent (NOK). In the third figure, it is dangling (NOK).



Figure 2.5: In the first figure, the hose is seated on the tub and the clamp is secure (OK). Next, it is occluded by an object and it cannot be inspected (NOK).

all the way secure as shown in figure 2.5 (to avoid water leakage).

EMI Filter The EMI (Electromagnetic interference) filter is a device that reduce the high frequency (radio frequency) noises present on the power lines that can degrade the performance of the electronic equipment inside the machine. It is inspected to make sure it is completely fastened down and is not loose as shown in figure 2.6.

Pressure Sensor Washing machines use pressure sensors to detect the water level in the tub. This information is used by the machine to estimate the amount of water



Figure 2.6: In the first figure, the EMI filter is completely fastened down (OK). Next, it is not fastened down completely (NOK).



Figure 2.7: In the first figure, the white connector of the pressure sensor is snapped in completely inside the blue mount (OK). Next, the part is occluded or it is a trigger failure (NOK).

needed for different cleaning cycles. The pressure sensing module is inspected to make sure it is snapped in completely to its mount as shown in figure 2.7.

Wire Holder The wire holder is designed to hold all the wire harnesses and avoid any dangling wires inside the machine. A wire harness is a group of wires that transmit power and other signals. It is important to manage the wiring so that wires are not left dangling around. Dangling wires have a risk of breaking or getting shorted and that might affect the performance of the washing machine, or it might just stop working due to no power.

The wire holder is inspected to make sure the wire harness is inside the wire



Figure 2.8: In the first figure, the wire harness is inside the wire holder (OK). Next, the wire holder is not tied and the wire is left dangling outside of the holder (NOK). Next, the holder is not tied and the wire can slip out (NOK). In the fourth figure, the wire is not going through the holder and in the last figure the holder is absent (NOKs).

holder and the holder is completely snapped into the frame of the machine. It is also inspected to make sure that it has a knot, so as to not allow the wire harness to slip out of the holder.

2.1.2 The thermistor data set

This data was collected to demo our system to the Samsung executives. We collected a total of 23,247 frames of machines, approximately 2 frames per machine. Two parts were inspected in each frame: the thermistor screws and the screw near

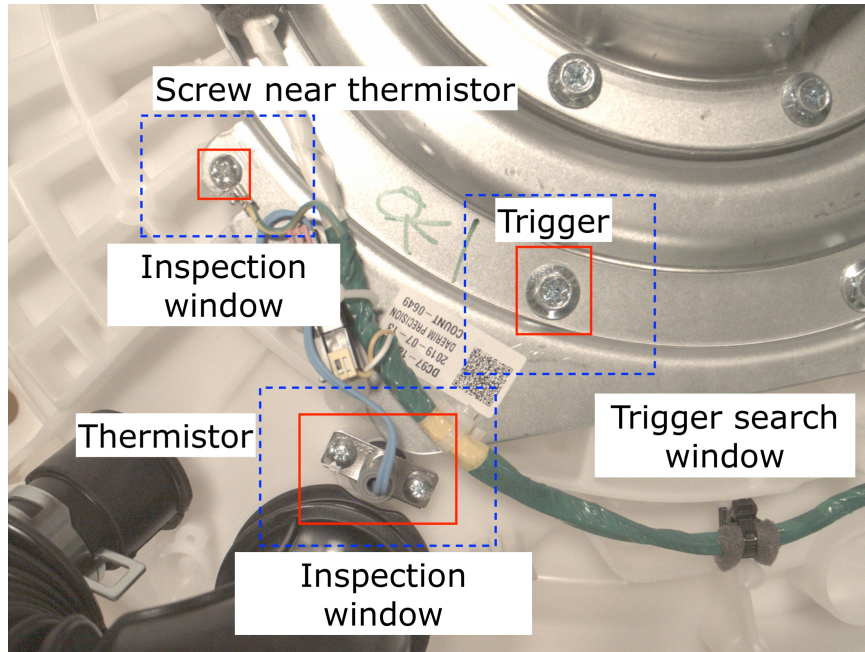


Figure 2.9: Thermistor data set: A frame of the tub of the machine; Two different parts inspected in 23,247 such frames.

the thermistor as shown in figure 2.9.

Thermistor screws A thermistor is a resistor, the resistance value of which, changes based on the temperature of the surroundings. Thus it acts as a temperature sensor. It is inserted inside the tub (where the clothes get washed) to measure the temperature of the water.

The thermistor is attached to a frame with 2 screws. The thermistor is inserted into the tub and the two screws are tightened to hold it in place inside the tub. Loose screws can cause water leakage, damaging other electrical components of the machine. Thus they are inspected to make sure the silver frame is fully seated and screwed into the tub as shown in figure 2.10.

Screw near thermistor The screw near the thermistor helps with some of the electrical connections. It is inspected to make sure it is present and screwed tightly



Figure 2.10: In the first figure, the thermistor frame is screwed into the tub completely (OK). In the second figure, there is a human hand covering the part and we cannot inspect it (NOK).



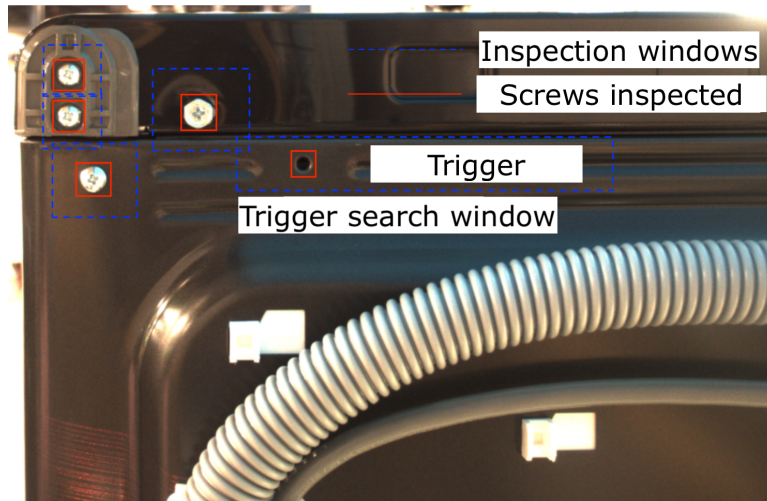
Figure 2.11: In the first figure, the screw is screwed tightly on to the frame (OK). In the second figure, the screw is absent (NOK).

into the silver frame as shown in figure 2.11

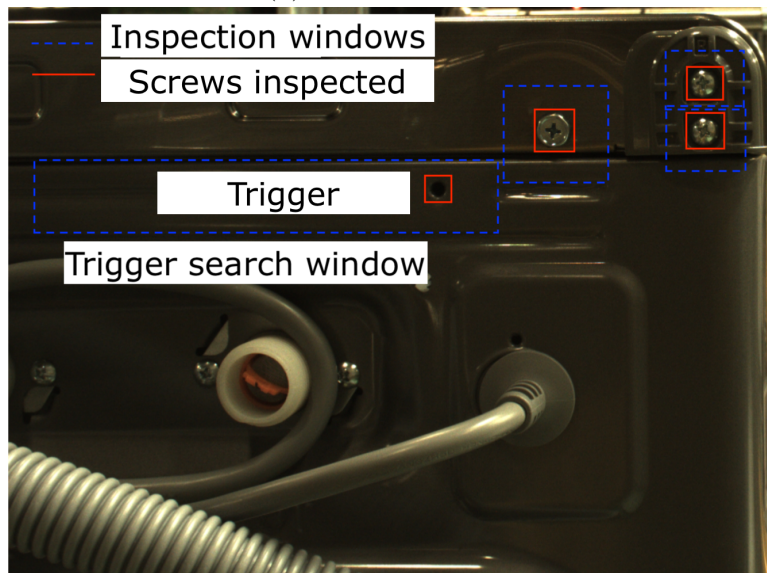
2.1.3 The SEHA data set

The SEHA data set was provided to us by the Samsung Electronics Home Appliances (SEHA) factory and contains 3987 frames of the screws on the back of the machine, on the left and right side as shown in figure 2.12. Out of the 7 screws, the one on the top left side and next to the grey colored frame (right side of the frame) was used for our experiments.

Top left screw on the back The top left screw was inspected to make sure it is tight and screwed completely into the frame as shown in figure 2.13.



(a) Top Left Screws



(b) Top Right Screws

Figure 2.12: Data Collection 3: Top left and top right corners of the back of the machine; Four screws at the top left corner and three screws at the top right corner inspected.



Figure 2.13: Screws on the back of the washing machine. The first figure is an OK screw. Next, the screw is absent (NOK). In the last figure, the screw is not screwed into the frame (NOK).

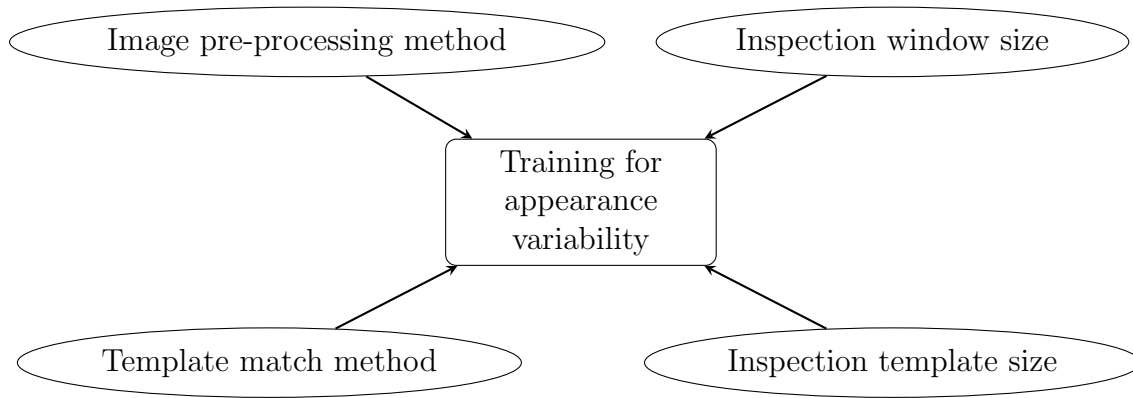


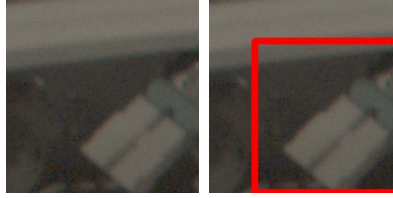
Figure 2.14: Different pre-processing options developed to improve the results of the clustering algorithm and other methods.

2.2 Pre-processing

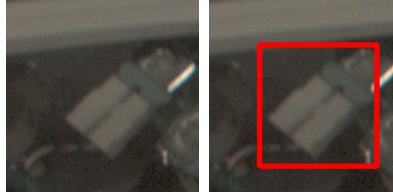
Figure 2.14 shows the several pre-processing options we developed, while trying to improve the results of our methods. Some helped clarify part appearances, while others improved part comparisons. Image pre-processing methods like histogram equalization and median blurring were used to improve contrast and remove noises in the images. Different template match methods like RGB or Edge matching were tried, to improve part comparisons while clustering. Other options developed were changing the inspection template size and the inspection window sizes, in order to improve the localization of the areas of inspection. Each of these pre-processing options are discussed in this section.

2.2.1 Inspection window size

Sometimes the part to be inspected is not centered or partially visible or clipped inside the inspection window. To solve this problem, the size of the inspection window is increased and all the inspection windows are re-cropped. The inspection windows are cropped in such a way that there are no nearby parts or other similar



(a) Inspection search window too small as part is not fully visible or clipped and the localization is poor.



(b) New inspection window with an increased size leads to better localization.

Figure 2.15: The system has an option to adjust the window size if the part is not fully visible in the cropped images.

objects in view. An example is shown in figure 2.15.

The inspection window for the white water valve was cropped originally small as shown in figure 2.15a. For a couple of thousand images the valve was at the right bottom and some part of the green mount was out of the frame. To solve this problem the size of the window was increased and all the inspection windows were re-cropped. Now, the templates were able to latch on to the correct location as shown in figure 2.15b.

2.2.2 Size of the inspection template

In our experiments, we consider the following factors while deciding the size of the template:

Area of inspection The size of the inspection template is chosen such that it covers the entire area of inspection, at minimum. Many times, the size of the template is increased to include any nearby context that might act as a cue for defects or might

help the template matching algorithm to locate the area of inspection easily (have a good template match image).

Appearance of defects Defects are looked at to get a better understanding of the differences between defects and OK parts. This information is used to adjust the size of the template so that it is able to capture defects and not just OK parts.

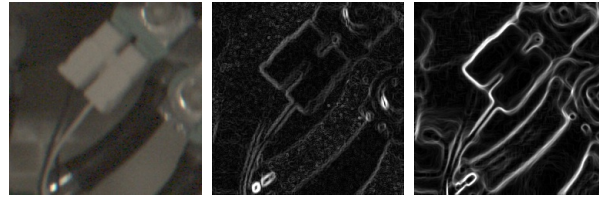
Variation in size All the cropped parts are checked for variations in the size of the area of inspection. Sometimes a part appears larger due to a change in camera angle or change in orientation of the assembly, other times, a part could have variable sizes, both cases need to be considered.

Image gradients Another important factor to consider is the image gradients. The image gradients are looked at and analyzed for defect and OK images. Sometimes they act as cues for defects, other times they help in the process of localization (edges of the nearby context).

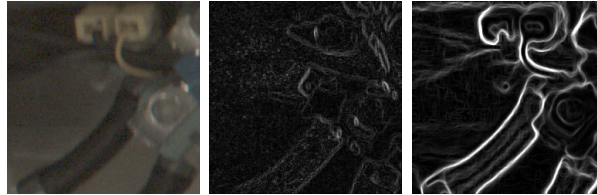
2.2.3 Image pre-processing method

For the clustering to work, it is important for the part to be clearly visible inside the inspection window. Image processing techniques like histogram equalization and median blurring are used to solve this problem. Histogram equalization improves contrast and median blurring removes the noises in the image. This can be seen in figure 2.16 for the white water valve use case.

That being said, it is not always necessary to use image pre-processing. Sometimes the lighting conditions are good and the image pre-processing only hurts, it does not improve the performance. Important edge information in the area of inspection



(a) Inspection window of an OK white water valve and its edge image before and after image pre-processing.



(b) Inspection window of an NOK white water valve and its edge image before and after image pre-processing.

Figure 2.16: Pre-processing drastically improves the edges of the parts which are dimly lit.

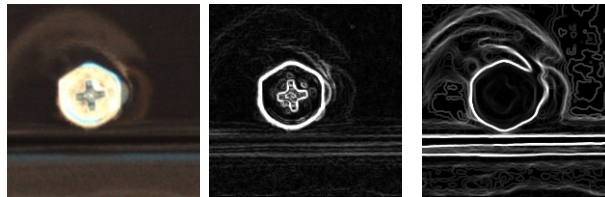


Figure 2.17: Pre-processing fails for this use case because the lighting is good and pre-processing is not required.

could be lost if pre-processing is used when not required, as shown in figure 2.17.

Table 2.2 gives information whether the image pre-processing option was activated or not, for all the 9 use cases. In our data sets, the screws were well lit and had strong edges, and hence, did not require image pre-processing to be done.

Inspection problem	Image pre-processing
White water valve	Yes
Red water valve	Yes
Tub clamp hose	Yes
EMI Filter	Yes
Pressure sensor	Yes
Wire holder	Yes
Thermistor screws	Yes
Screw near thermistor	No
Back screw	No

Table 2.2: Image pre-processing usage for different use cases depending on the strength of edges and lighting conditions.

2.2.4 Template matching method

The template matching algorithm has two different outputs, one is the match location and the other is the match score. These outputs could be generated using edge images or RGB images of the IT and IW, depending on the inspection problem.

Finding the match location Three factors are considered when deciding on the matching technique to be used for finding the match location: Distinctiveness of the edges in the area of inspection with respect to the nearby context, distinctiveness of the color of the area of inspection with respect to nearby context, color variation in the area of inspection and the nearby context and occlusions. The table 2.3 gives information regarding the matching technique used in each of the 9 use cases and why it was used.

Inspection problem	Match method	Notes
White water valve	Either	Part has distinct color and edges and no occlusions. Both RGB and edge matching work well.
Red water valve	RGB	Part has distinct color and edges but a lot of occlusions. Hence RGB matching (occlusions mess up edges)
Tub clamp hose	RGB	Edges of the area of inspection are weak and there are occlusions. Part has a distinct color.
EMI Filter	Either	Part has distinct color and edges. Both RGB and edge matching work well.
Pressure sensor	RGB	Part has distinct color and edges but a lot of occlusions. Hence RGB match.
Wire holder	Either	Part has distinct color and edges. Both work same.
Thermistor screws	RGB	Part has distinct color and edges but has occlusions. Hence RGB match.
Screw near thermistor	Edge	Part has a distinct circular edge. Hence edge match.
Back screw	Edge	Part has distinct color and edges and no occlusions. But there are too many color variations.

Table 2.3: Match methods used for different inspection problems.

Finding the match score For the parts inspected in this thesis, matching edge images of the ITs and the area of the inspection generate a wide range of scores and it is easy to decide on a threshold (score), below which, all the parts are labelled as defects. This is because the edges in an NOK or an image of a defective part are very different from those in an OK image.

Matching RGB images generate scores in a very narrow range and it is really difficult to decide on a threshold on such a narrow range of scores. Hence RGB matching is not used to score any of the inspection problems discussed in this thesis.

2.3 Ground truth

Once the inspection windows are cropped, they are manually reviewed or ground truth for defects. This helps us in understanding the appearance of the defects, and also helps us in evaluating the performance of the system in detecting defects, after the training process is completed.

The cropping process crops all the inspection windows of a part into one folder. These are then viewed using the icon view option of the macOS Finder (file explorer) as shown in figure 2.18. In the icon view option, one can see many inspection windows at a glance and that speeds up the reviewing process. As shown in the figure, 28 inspection windows of the white water valve were reviewed or ground truth at a time. It took on an average of 180 minutes to ground truth a data set of approximately 39,000 inspection windows for one use case.

Table 2.4 gives information regarding the number of images cropped and the number of defects found for all the 9 use cases. A total of 249,371 images were

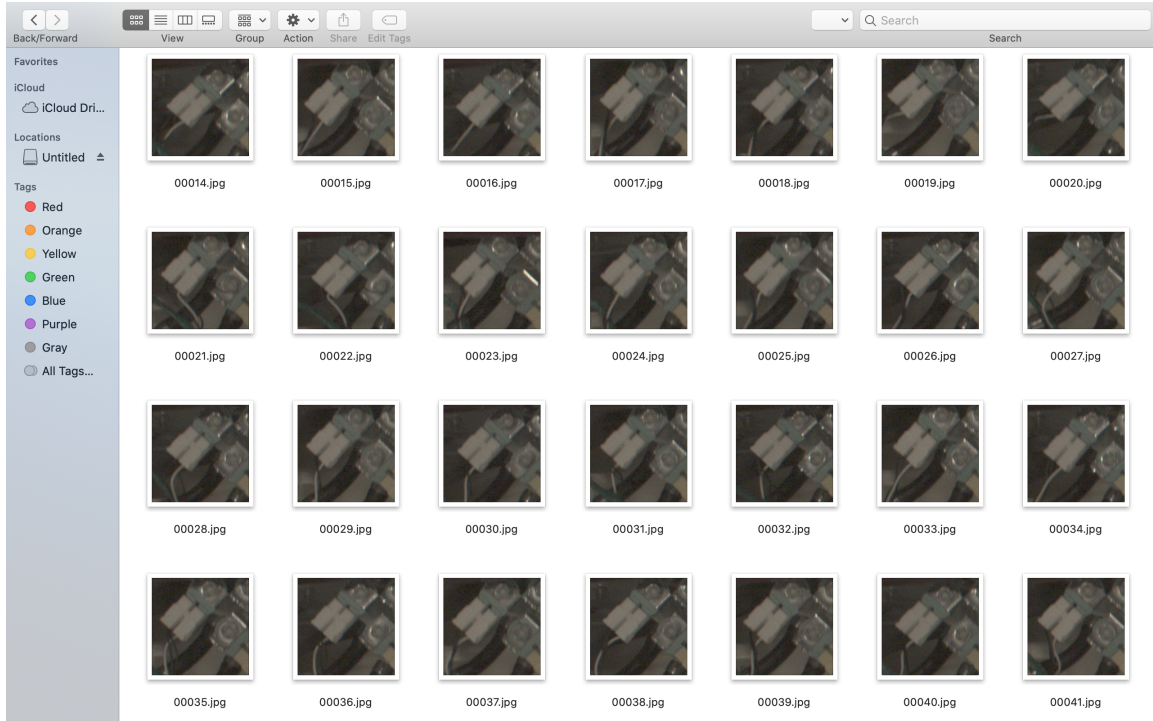


Figure 2.18: Process of manual reviewing for defects.

cropped for 9 inspection problems and there were 2 to 180 defects per inspection problem. Some of the images failed to crop due to trigger failures (failed image triggering) or out of bound inspections (part not in frame).

2.4 Clustering

For this thesis, the clustering algorithm is modified to process a much larger amount of data. The method now has very less memory requirements and can run on arbitrary sized data faster.

The implementation in [7] loads all the inspection windows of the part from the database in order to run clustering. The size of an inspection window can be of

Inspection problem	No. of frames	Frames analyzed	Fail to crop	Defects	Defect rate
White water valve	39,268	32,242	1,602	83	0.26 %
Red water valve	39,268	38,447	821	89	0.23 %
Tub clamp hose	39,268	39,268	0	17	0.04 %
EMI Filter	39,268	39,268	0	31	0.08%
Pressure Sensor	39,268	13,009	26,259	2	0.02 %
Wire holder	39,268	36,658	2,610	180	0.49 %
Thermistor screws	23,247	23,247	0	12	0.05 %
Screw near thermistor	23,247	23,245	2	23	0.1 %
Back screw	3,987	3,987	0	39	1.0 %

Table 2.4: Crops and ground truths for all the use cases.

several megabytes and thus the program might require several gigabytes of memory to run. If we have hundreds of thousands of images then the memory requirement can go up to several terabytes. Considering the time complexity of the clustering algorithm of $O(N^2)$, that means more memory usage and more time to run. In contrast, the implementation in this thesis requires very less memory and runs faster for large data sets. At any given point during which the program runs, there are only two images that are loaded. Thus the memory requirement to run the program is only a few megabytes. This updated version can now handle arbitrary sized data unlike the previous implementation, where if the data size was greater than the available memory, the program would crash because it couldn't load all the images.

2.5 Training and evaluation

Figure 2.19 gives an overview of the training and evaluation process used to train an inspection system prototype. We begin the training process by first looking at the output generated by the clustering algorithm, viz., the cluster seeds and their members. An application called cluster-viewer is used to view these. Next, depending on a few properties of the output clusters a decision is made whether to tweak the pre-processing to run clustering again, or, to input a set of seeds as training templates to the inspector. A program called seed-viewer helps in selecting the seeds to be input as training templates to the inspector. The inspector runs an inspection system simulation on the data that has been ground truth using the training templates. The performance of this simulation is evaluated by the evaluator. An operating point set is using ROC curves, for which the defect detection rate is high and the false alarms per defect rate is low. The inspection simulation is re-run for more templates or the problem is converted to a 2-class problem if it is not possible to set an operating point that separates the OKs from the NOKs. Each of these processes are explained in detail in this section.

2.5.1 Cluster-viewer

The clustering algorithm outputs different cluster seeds and its members in the form of a text file. A program called cluster-viewer is used to view the different cluster seeds and its members along with their match locations and match scores, inside their respective inspection windows. The cluster seeds are expected to capture the different appearances of the parts, including defects, and the cluster members are expected to look like their respective cluster seeds.

By default, the number of clusters generated is set to 20 and is increased if

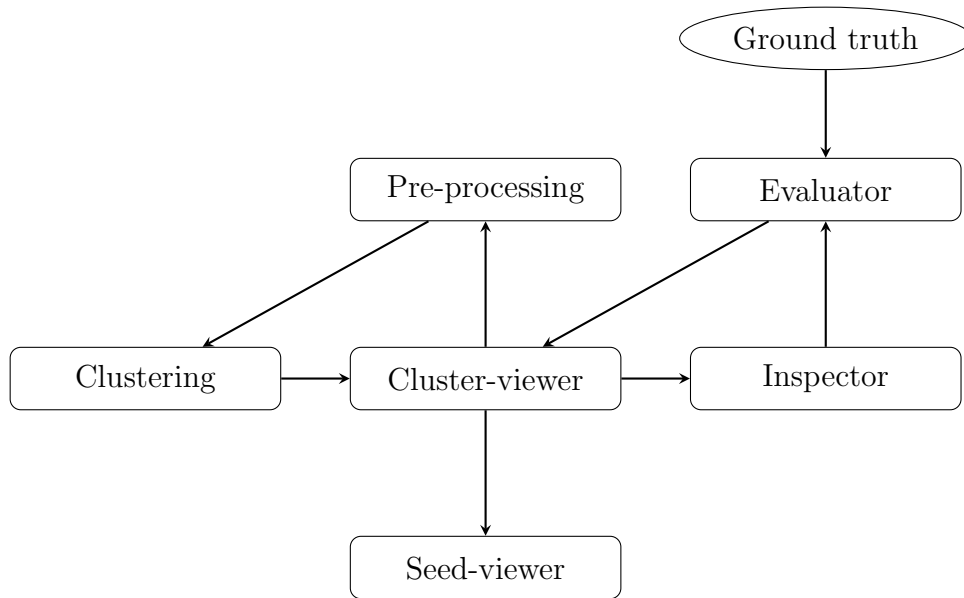


Figure 2.19: Overview of the methods used for training and evaluation of an inspection system prototype, for the learned variability.

the performance of our method for 20 clusters is not satisfactory. Three things are checked at this point: localization of the parts inside the inspection windows, defect clusters and inter cluster appearance variability.

If the parts are not localized inside their inspection windows, the reason for the bad localization is diagnosed by looking at the gradient images of the cluster seeds in each of the clusters. The reason we look at gradient images is because we do all the scoring using gradients and clusters are formed depending on these scores. Accordingly, the pre-processing is tweaked and the clustering algorithm is re-run.

We also look at the defect cluster seeds and its members. Ideally, the number of members in all the defect clusters should add up to the number of defects found during ground truth. If the defect clusters have a lot of OK seeds, that indicates that the clustering was not able to separate out all the different appearances. The OK seeds inside defect clusters are probable false alarms and one of our goals in training the inspection system is to reduce these false alarms. Hence, in this case, clustering

is run for more number of clusters with the hope that these OK parts hiding in defect clusters might come out as separate clusters or join other OK clusters.

Sometimes, even if a part has a lot of defects, there are no defect clusters formed. This is a clear indication that more clusters need to be generated; clustering needs to re-run for more clusters.

2.5.2 Seed-viewer

Once satisfactory results are generated, meaning, the clustering is able to generate seeds that have parts localized well and capture the different appearance variabilites along with defects, a set of OK seeds are input to the inspector as training templates. A program called seed-viewer helps in viewing all the seeds at a glance so that it is easy to decide the training templates.

Figure 2.20 shows the output generated by the seed viewer for the EMI filter. It can be seen that there are OK templates as well as NOK templates suggested by the clustering algorithm. A set of these OK seeds, or a set of both, OKs and NOKs (for a 2 class problem) would be input as training templates to the inspector.

2.5.3 Inspector

The inspector, as the name suggests, inspects, or in this case, runs an inspection system simulation for the training templates selected. The inspector compares these templates with each and every inspection window in the ground truth data set. For each IW in the data set, one of the template matches with a higher score than other templates. The name of the IW, the highest match score and the respective template are saved in a text file as shown in table 2.5.

Depending on the complexity of the inspection problem, the inspection prob-

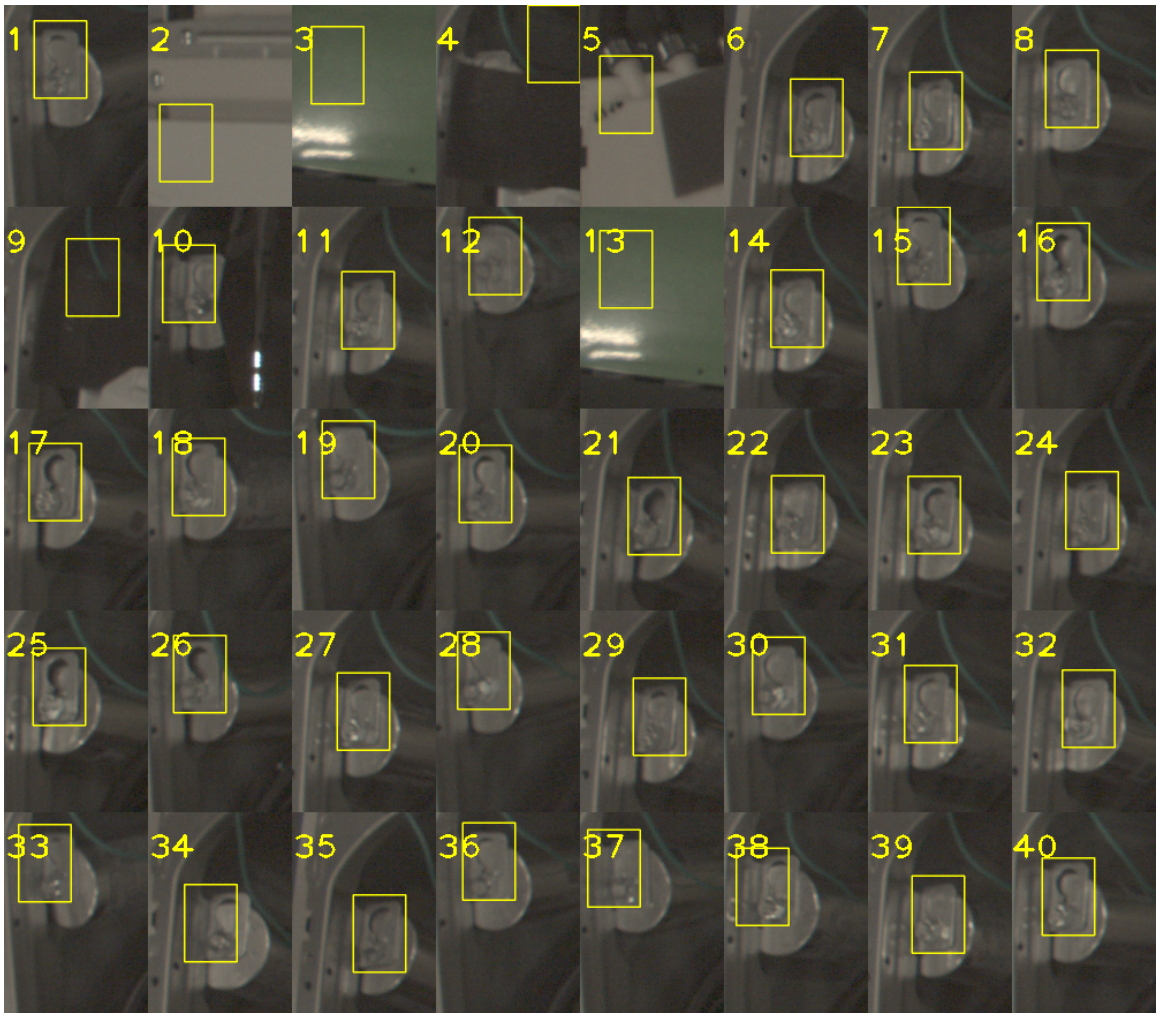


Figure 2.20: The 40 cluster seeds capture different variations in the appearance of the EMI filter, including defects.

lem could be configured as one of the two types:

1 class inspection problem It is expected that all inspection windows with OK parts would match with good scores and all the defects would match with low scores. The OK seeds of large OK clusters are selected as training templates. In this case, the problem is a 1 class problem where we are comparing only OK training templates and the low scoring parts are expected to be defects.

2 class inspection problem For some inspection problems like the one shown in figure 2.20 the defects and the OK templates look the same. It is difficult, even for a human eye, to spot differences that differentiate an OK from an NOK. The edge images of the defects and non defects look similar and have match scores in a similar range. In such cases the defects may or may not score low. The inspection problem is then converted to a 2 class inspection problem, where we compare all the inspection windows with OK templates and then with NOK templates and label each inspection window as belonging to one of the classes depending on which template scores higher. In this case, the OK seeds of large OK clusters are selected as class 1 and the NOK seeds of large NOK clusters are selected as class 2.

2.5.4 Evaluator

The evaluator, as the name suggests, is used to evaluate the performance of our inspection system simulation performed by the inspector. The inspector output is compared with the ground truth to evaluate its performance in detecting defects. The goal is to decide an operating point or the score (threshold) that separates the OKs from the NOKs (for a 1 class problem), or decide two different operating points,

Name of the part inspected	Score	Template that matched	Ground truth
00042.jpg	0.71	2	OK
00043.jpg	0.77	0	OK
00044.jpg	0.70	1	OK
00045.jpg	0.72	1	OK
00046.jpg	0.82	1	OK
00047.jpg	0.77	0	OK
00048.jpg	0.73	1	OK

Table 2.5: A glimpse of the inspector output for the red water valve. The inspector stores the name of the image, maximum match score, the template that matched with that score and the ground truth label. In this case there were 4 templates numbered from 0 to 3.

one for each of the classes (for a 2 class problem) (for the purpose of explanation we are considering a 1 class problem here). The following metrics are used to decide the threshold:

Defect detection rate: The defect detection rate of a part is the amount of defects detected or true negatives out of the total number of defects. It can be defined as:

$$\begin{aligned}
 \text{Defect detection rate} &= \frac{\text{Number of defects detected}}{\text{Total number of defects}} \\
 &= \frac{\text{True negatives}}{\text{True negatives} + \text{False positives}}
 \end{aligned} \tag{2.1}$$

False alarms per defect: The false alarm rate of a part is the amount of false alarms or false negatives out of the total number of defects

$$\begin{aligned}
 \text{False alarm rate} &= \frac{\text{Number of false alarms}}{\text{Total number of defects}} \\
 &= \frac{\text{False negatives}}{\text{True negatives} + \text{False positives}}
 \end{aligned} \tag{2.2}$$

Index	Name of the part inspected	Score	Ground truth
1	00195.jpg	0.43	NOK
50	22250.jpg	0.57	NOK
100	22246.jpg	0.60	NOK
150	22988.jpg	0.61	OK
200	22081.jpg	0.62	OK
300	01210.jpg	0.63	OK
500	04082.jpg	0.64	OK

Table 2.6: A glimpse of the inspector output sorted in ascending order of the match scores for the red water valve for 4 templates, shown for the 1st least scoring part to the 500th least scoring part.

Setting the threshold using ROC curves The first step in setting the threshold would be to sort all the part images inspected, in ascending order of their match scores, as shown in table 2.6. Now, the least scoring N parts are going to be labelled as defects by the inspector. In order to decide the operating point or the threshold, the metrics discussed earlier would have to be calculated for N low scoring parts, where N is arbitrary and should be at least equal to the number of defects. Thus we would have to calculate the defect detection rate and the false alarm rate for 1 least scoring part, 2 least scoring parts, 3 least scoring parts, 4, ..., N least scoring parts until we get a close to 1.0 defect detection rate and a low false alarm rate (we try to keep it under 2.0 for most parts) and the threshold value would be the score at this Nth least scoring part. This is a tedious process and it is better to plot the data and decide the threshold. Hence we use ROC curves.

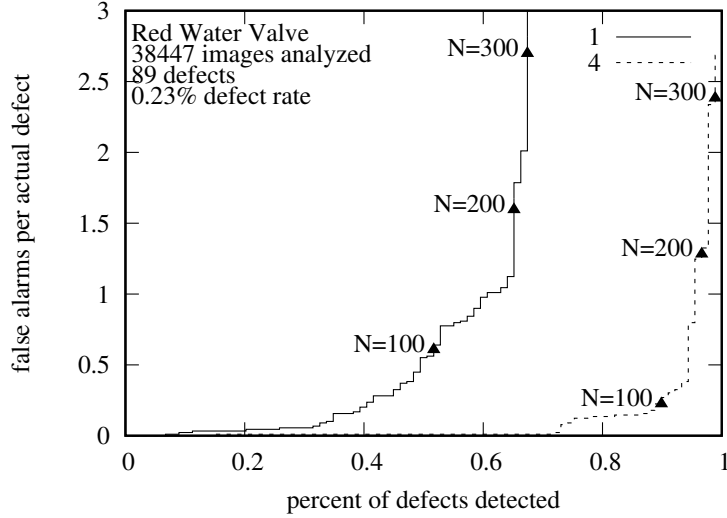
The receiver operating characteristic (ROC) curves are plotted for different values of N least scoring parts. The sorted inspected output shown in figure 2.6 is

N least scoring parts	Actual defects	False alarms	Defect detection rate	False alarm rate	Score of the Nth part (threshold)
50	49	1	0.55	0.01	0.57
100	80	20	0.89	0.22	0.60
150	84	66	0.94	0.74	0.61
200	86	114	0.97	1.28	0.62
300	88	212	0.99	2.38	0.63

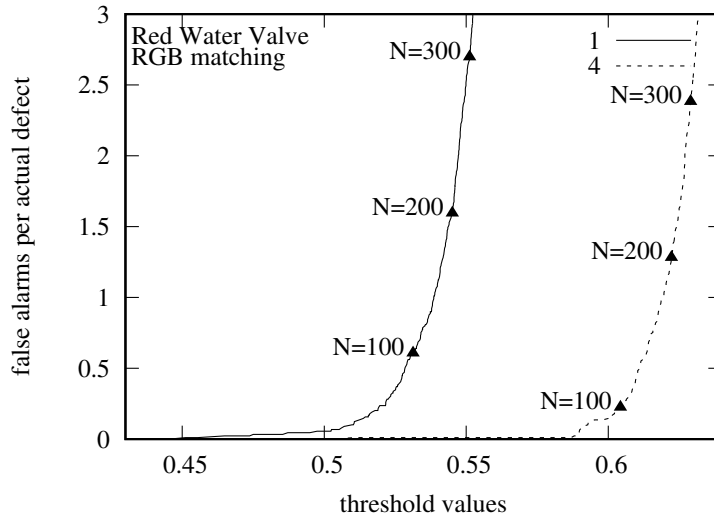
Table 2.7: Values of different metrics, used for drawing the ROC curves for 4 red water valve training templates.

used to generate another text output that is used to generate the ROC curves. This is shown in table 2.7 for the red water valve. The first column in this table is the number of least scoring members (N) considered. The second and third columns are the number of defects and false alarms for the N least scoring members. The fourth and fifth columns are the defect detection rates and the false alarm rates and in the final column is the threshold or the score of the Nth least scoring member. Thus the threshold value would be the score of the Nth least scoring part at which the N least scoring parts would have a high defect detection rate and a low false alarm rate. Thus the ROC curves are basically, the defect detection rate, the false alarm rate and the Nth least score plotted for different values of N. As we have three different values that need to be plotted, we plot two kinds of ROC curves. One is used to evaluate the performance and the other is used to set the threshold. Both are plotted for different values of N.

The curve shown in figure 2.21a is used to evaluate the performance of the simulation. It has the defect detection rate on the X axis and the false alarm rate on the Y axis plotted for different values of N. In this case they are plotted for 1 least score to 500 least scores. As expected, as you go on increasing the value of N more



(a) False alarm rate vs defect detection rate



(b) False alarm rate vs threshold values

Figure 2.21: ROC curves of the red water valve for 1 training template and 4 training templates drawn for 1 least scoring part to 500 least scoring parts. For 4 training templates, the operating point is at the right bottom corner with a high defect detection rate and a low false alarm rate. The scores in this region are in the range of 0.58 to 0.62 and the threshold can be set to one of those values for inspecting this part.

defects are found. In this case we have drawn the curves using 1 training template and another curve using 4 training templates of the red water valve. It can be seen that 4 training templates are able to find all the defects at a false alarm rate of around 2.5 with around 300 (N) least scoring parts considered.

The curve in figure 2.21b helps to set a threshold that separates the OKs from the NOKs. It has the Nth least score (threshold) on the X axis and the false alarm rate on the Y axis plotted for different values of N. It is set in a such a way that the false alarm rate is low and the corresponding defect detection rate in figure 2.21a is high.

Ideally, for a certain threshold, the false alarm rate should as low as possible (less than 2) and the defect detection rate should be as high as possible (close to 1). Such a system would be able to detect all the defective machine parts or inspection problems and for the most part (50 % of the time), raise an alarm only if there is a defect. We typically choose the knee of the curve as the threshold value. The knee of the curve is where the curve bends from a low slop to a high slope. In other words the knee is the point after which the false alarm rate increases and there is not much significant, or zero increase in the defect detection rate. In this case, for 4 templates, the knee is at a threshold value of 0.6. The false alarm rate at this value is around 0.25 and for a false alarm rate of 0.25 we have a defect detection rate of around 0.90 in figure 2.21a, which is not bad.

There could be several iterations of clustering and running the inspector until the ROC curve in figure 2.21a has a knee or an operating point. Sometimes, even at higher values of N (much greater than the number of defects) the number of defects found are still low. In this example, for just 1 training template, the curve in figure 2.21b has a knee at around 0.53. For this value, the false alarm rate is around 0.25. At this false alarm rate, the defect detection rate in curve 2.21a is only around 0.4.

Thus only 40 % defects are detected. In this case, after increasing the number of training templates, the performance is seen to improve drastically.

Chapter 3

Results

This chapter describes how the inspection system prototype was trained for appearance variability, for all the 9 use cases. It also reports the number of defects detected and the number of false alarms generated by our inspection system prototype for each use case.

3.1 White water valve

The inspection template chosen for the white valve is shown in figure 3.1. It was seen that there is change in the intensity of color on all 4 sides of the part. Thus while cropping the template, 5 pixels were left from each side to make use of the color gradients, in order to make the part more distinct, and easier to latch on to, for the



Figure 3.1: The template chosen for the white valve.



Figure 3.2: OK templates suggested by the clustering algorithm for the white water valve, in decreasing order of their sizes.

template matching algorithm.

Inspector results The inspector was run for all the 4 OK templates (4 OK seeds with the biggest cluster sizes) suggested by the clustering algorithm and ROCs were generated. The ROC curves for the bottom 200 scores are shown in figure 3.3. The fourth template was omitted because it did not affect the performance in any way. It can be seen that we can get a defect detection rate of 1.0 with 0.0 false alarms per defect.

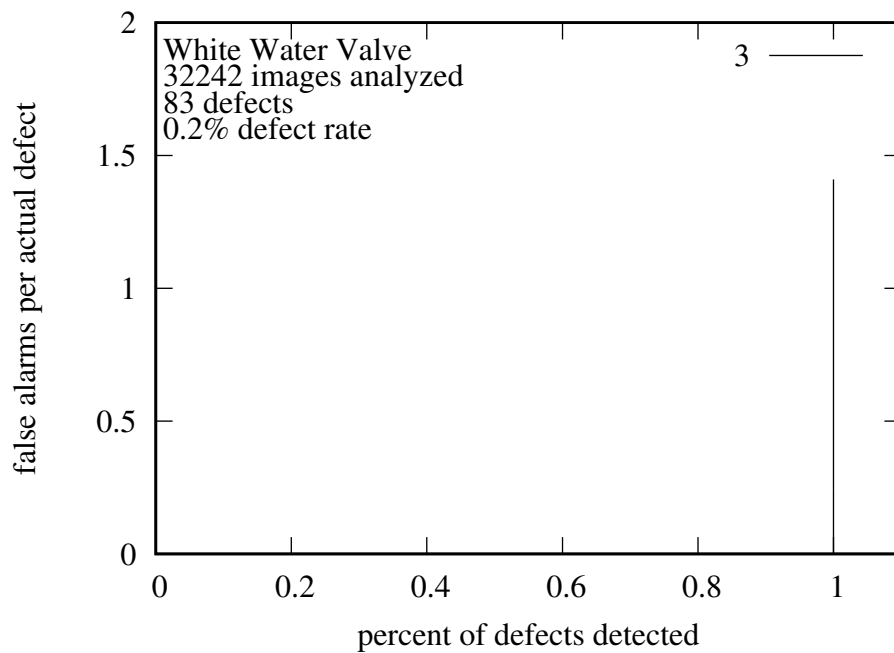


Figure 3.3: ROC curves of the white water valve for three templates. The operating point is at the right bottom corner with a 1.0 defect detection rate and a 0.0 false alarm rate.



Figure 3.4: The template chosen for the red water valve.



Figure 3.5: OK templates suggested by the clustering algorithm for the red water valve in decreasing order of their sizes.

3.2 Red water valve

The first seed or the inspection template given as input to the clustering algorithm is shown in figure 3.4. It can be seen that it is cropped in a way that leaves 10 pixels from each side of the inspection area. A similar approach was taken to decide the size of the template as discussed for the white water valve.

Inspector results: Figure 3.5 shows the OK templates (OK seeds with biggest cluster sizes) suggested by the clustering algorithm for the red water valve. It can be seen that there is variability in the appearance due to the wire occluding the valve and due to the valve being tilted in some cases. The red valve sometimes appears smaller and that is captured by the clustering algorithm as seen in the second figure from left.

The inspector was run on all the 4 OK templates as suggested by the clustering

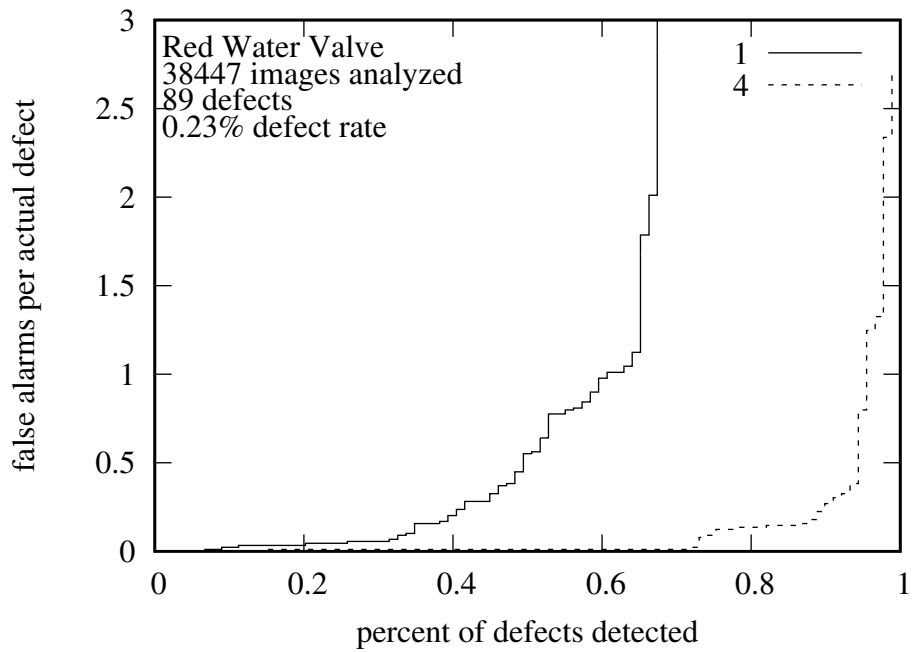


Figure 3.6: ROC curves of the red valve for one and four templates. The operating point is at the right bottom corner of the curve drawn with four templates, with a defect detection rate of 0.9 and a false alarm rate of 0.2.

algorithm. The ROC curves for the bottom 200 scores are shown in figure 3.6. It can be seen that we can get a defect detection rate of 1.0 with 2.5 false alarms per defect.

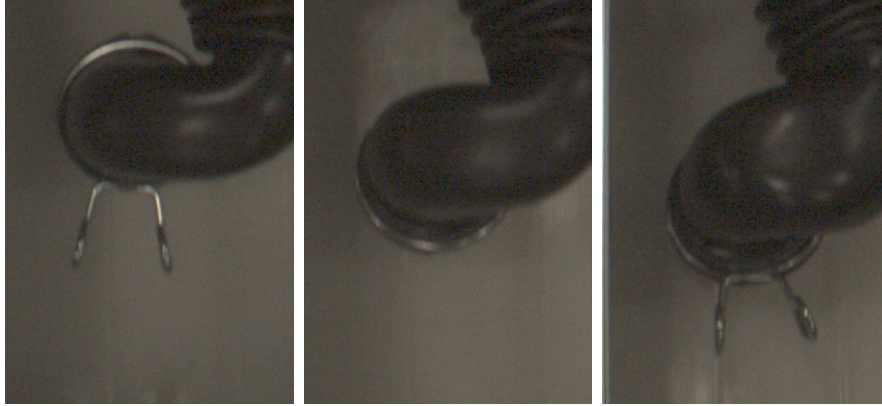


Figure 3.7: The handles of the clamps of the hose vary in their appearance and sometimes, are not visible at all.

3.3 Tub clamp hose

For this use case, we are inspecting both, the hose and the clamp. After looking at all the inspection windows of the hose with its clamp, it was seen that there was a lot of variance in the appearance of the area of inspection. The handles of the clamp would be at different locations and orientations around the hose as shown in figure 3.7.

We had three different options for templates as shown in figure 3.8. The second template is too large, as big as the inspection window size and it causes the template matching algorithm to fail. The third option would fail if the handles of the hose were occluded by the hose itself as shown in the second image in figure 3.7. Hence the area of inspection was reduced to just the hose and the ring around the hose as shown in figure 3.9. That was enough to detect whether the hose was seated and the clamp was secured.

It was seen that there is change in the intensity of color on 3 sides of the part. Thus while cropping the template, 5 pixels were left from each side to make use of the gradient information, in order to make the part more distinct, and easier to latch

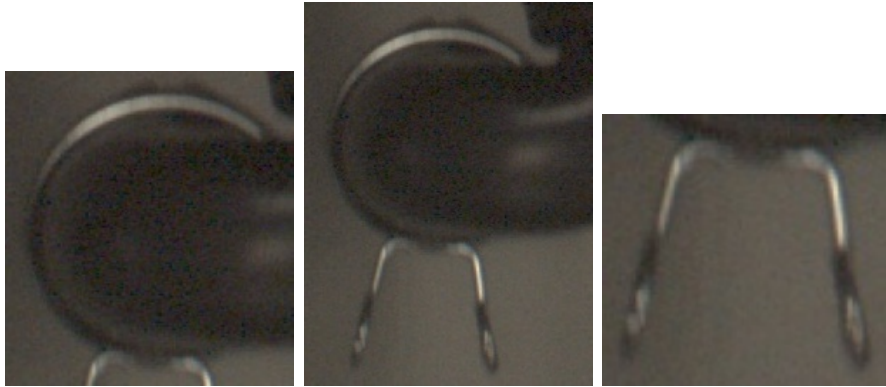


Figure 3.8: The different template options we had.



Figure 3.9: The template chosen for the tub hose with the clamp ring and the hose visible.

on to, for the template matching algorithm as shown in figure 3.9.

Inspector results Figure 3.10 shows the OK templates (seeds with the largest clusters) suggested by the clustering algorithm for the tub clamp hose. It can be seen that there is variation in how the clamp ring appears in each of the templates suggested by the clustering algorithm.

The inspector was run for 6 OK templates suggested by the clustering algorithm and ROCs were generated. The ROC curves for the bottom 200 scores are shown in figure 3.11. It can be seen that we can get a defect detection rate of 1.0 with 0.05 false alarms per defect.

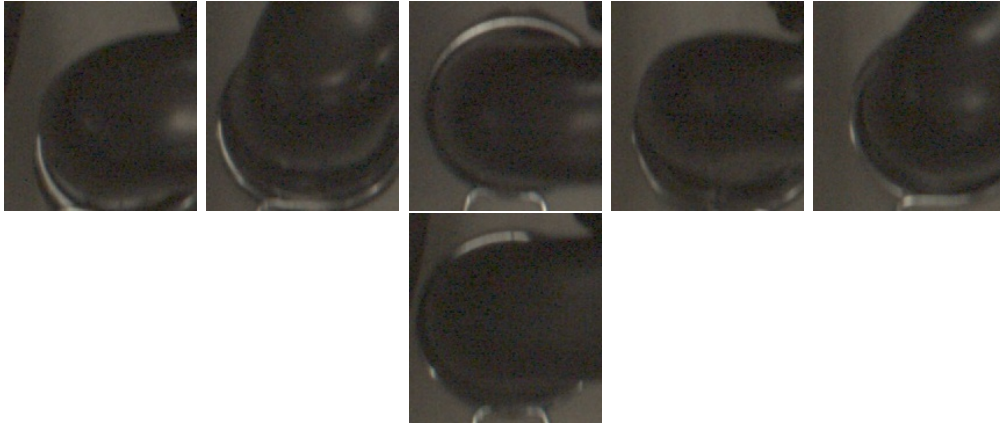


Figure 3.10: OK templates suggested by the clustering algorithm in decreasing order of their cluster sizes.

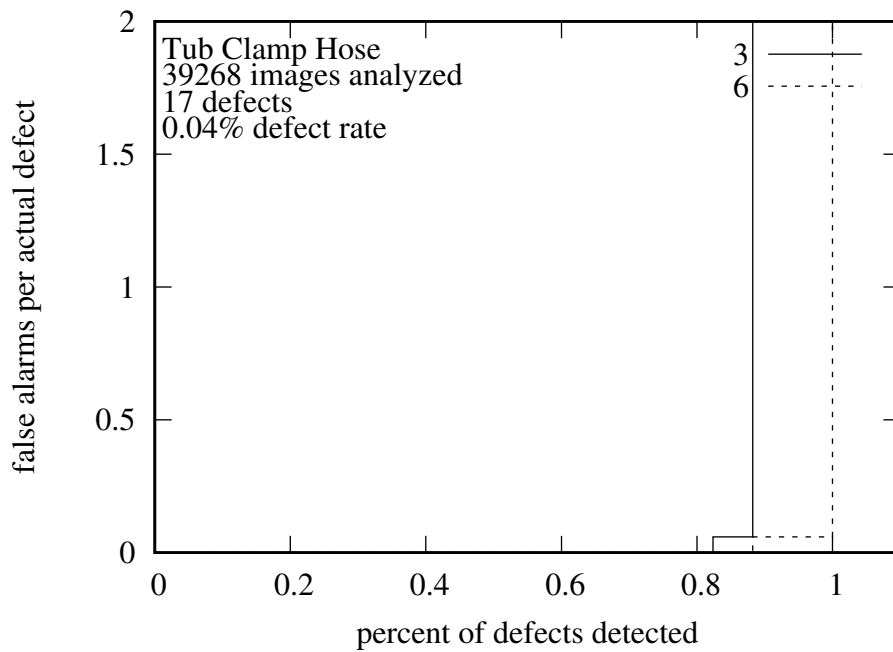


Figure 3.11: ROC curves of the hose for three templates and six templates. The operating point is at the right bottom corner for the curve drawn with six templates, with a defect detection rate of 1.0 and a false alarm rate of 0.05.

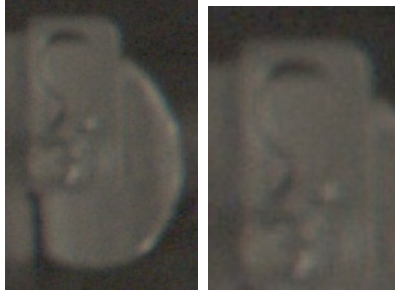


Figure 3.12: Different options we had for choosing the template for the EMI filter.



Figure 3.13: An OK and an NOK image of the EMI filter.

3.4 EMI Filter

As shown in figure 3.12 we had two options when choosing the template for the EMI filter. As seen in figure 3.13, the difference between an OK image and an NOK image was noticeable in the area where the part was fastened down to the hook, while the overall shape remained the same. If we chose the first template image, a defect image would match the template with a reasonably high score because it matched

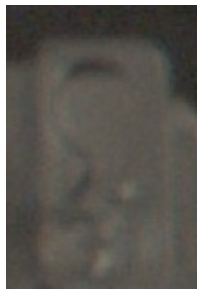


Figure 3.14: The template chosen for the EMI filter.



Figure 3.15: OK seeds of the EMI filter suggested by the clustering algorithm in decreasing order of their cluster sizes.



Figure 3.16: NOK seeds of the EMI filter suggested by the clustering algorithm in decreasing order of their cluster sizes.

other parts of the template like the semicircular arc and the edge at the top of the part. Hence we decided to restrict the template size to the area of inspection, which is the area of the frame onto which the part was hooked on, as shown in figure 3.14.

Inspector results: Figure 3.15 shows the OK training templates, while figure 3.16 shows the NOK training templates suggested by the clustering algorithm and used by the inspector to inspect for defects. In both the cases the variability in the appearance due to change in viewing angle is accurately captured by the clustering algorithm.

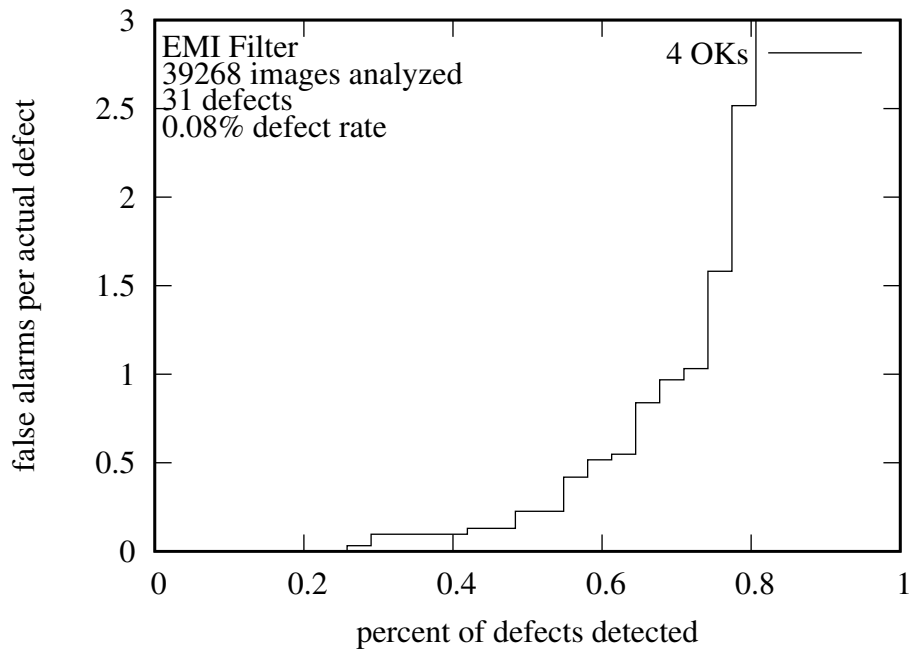


Figure 3.17: EMI filter as a 1 class problem using 4 OK training templates detects around 80 % defects with a false alarm rate of 3.0.

The EMI filter problem was first solved as a 1 class problem as shown in figure 3.17 and 3.18, for an OK class, and then for an NOK class respectively. The least scoring parts were considered while drawing the curve for the OK class, while for the NOK class, the highest scoring members that matched the defect templates were considered.

After manually comparing the defects found by the OK class and the NOK class, the results were combined as shown in figure 3.19. The 2 class system has a defect detection rate of 0.97 with 2.9 false alarms per defect, better than the 1 class system.

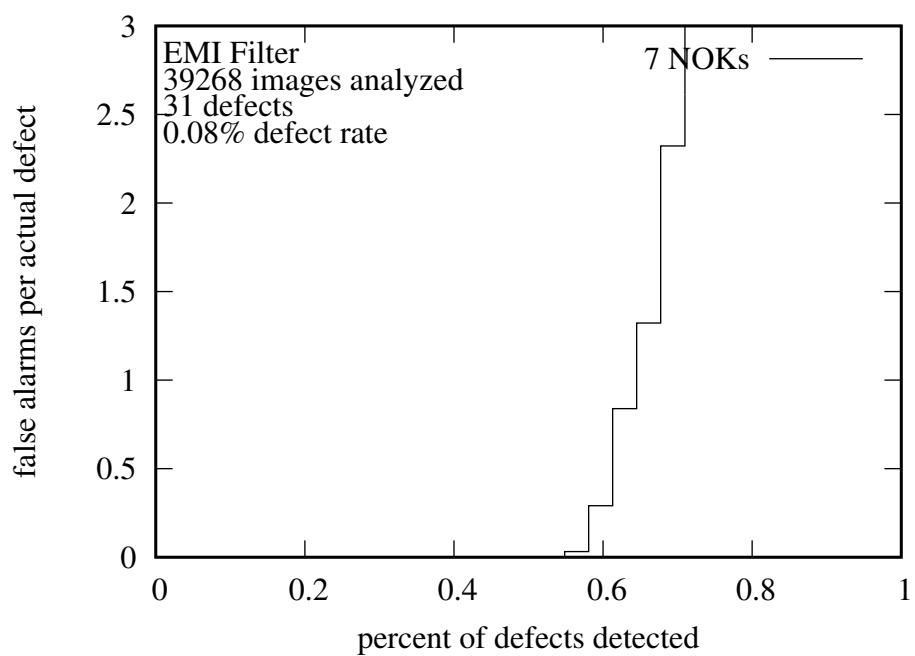


Figure 3.18: EMI filter as a 1 class problem using 7 NOK training templates detects around 70 % defects with a false alarm rate of 3.0.

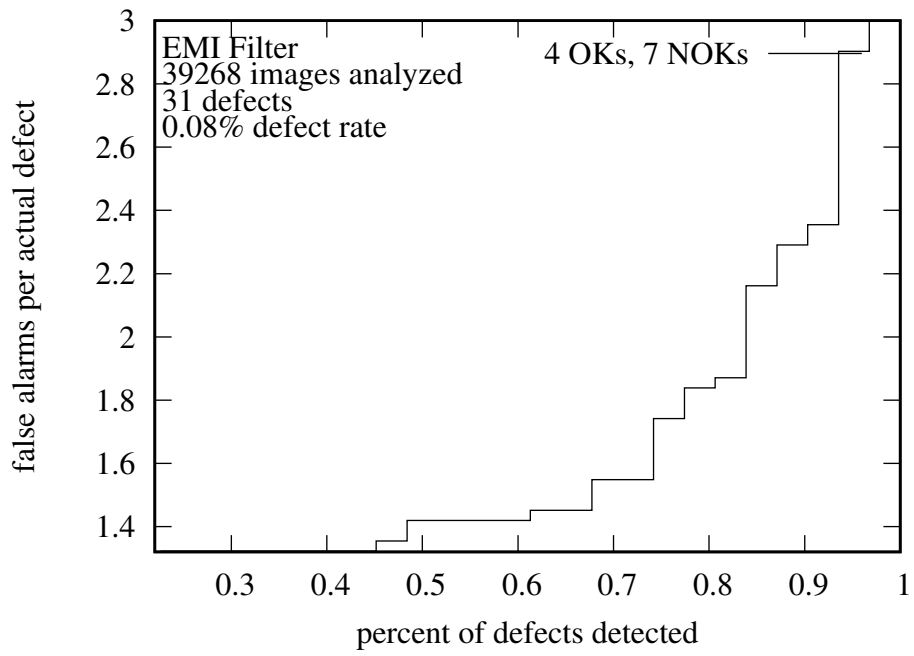


Figure 3.19: EMI filter as a 2 class problem combining the results from 4 OK and 7 NOK templates has a peak defect detection rate of 0.97 with 2.9 false alarms per defect.



Figure 3.20: The template chosen for the pressure sensor.

3.5 Pressure Sensor

A procedure similar to the white water valve was followed to select the template for the pressure sensor as seen in figure 3.20.

Inspector results The pressure sensor data has only 2 defects and the template input to the clustering algorithm was able to detect both the defects with 0 false alarms. This is because the number of images is very small for this part and just one template was enough for the system to detect both the templates, even though the part has a lot of variability in its appearance.

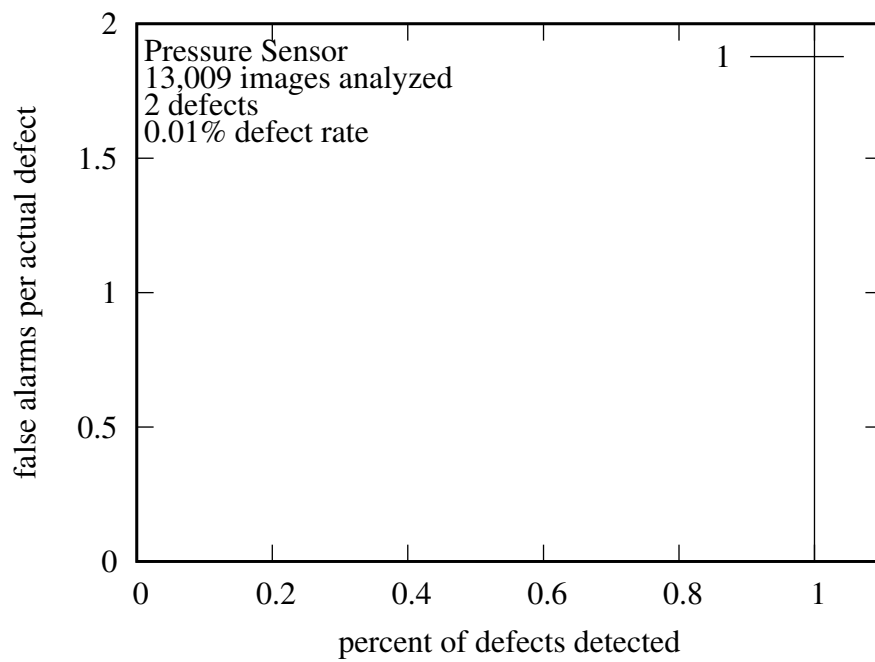


Figure 3.21: ROC for the pressure sensor. With just 1 template the system was able to detect all the defects with 0 false alarms.



Figure 3.22: The template chosen for the wire holder.

3.6 Wire holder

Figure 3.22 shows the inspection template chosen for the wire holder. This was the hardest inspection problem and the inspection system could find only 50 % of the defects at a high false alarm rate of 5.0 as shown in figure 3.23.

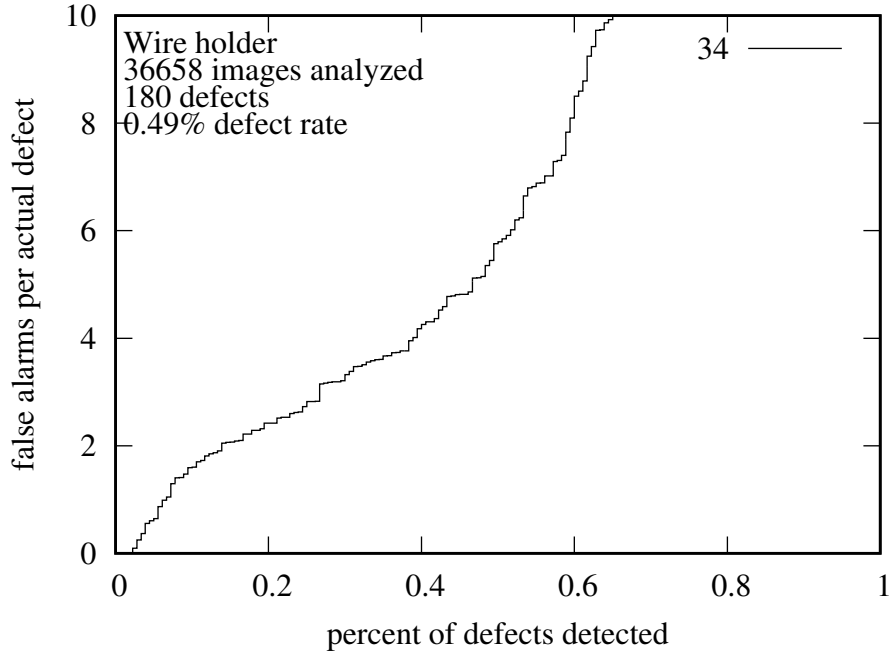


Figure 3.23: ROC for the wire holder using 34 training templates. The system was able to find only around 50 % of the defects at a high false alarm rate of 5.0.



Figure 3.24: The thermistor screws appear larger if closer to the camera.

3.7 Thermistor screws

The thermistor screws are at the back of the tub and the tub is raised when it reaches the work station where a human engineer assembles another part on it. This causes the frames of the part to be captured at different heights and thus the thermistor screws appear larger in some frames as shown in the image on right in figure 3.24.

In order to take the change in appearance into account, the inspection template



Figure 3.25: The template chosen for the thermistor screws.

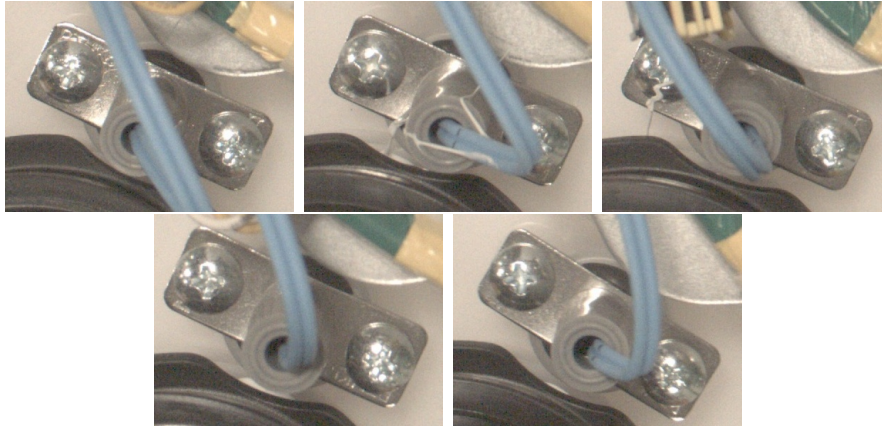


Figure 3.26: OK seeds of the thermistor screws suggested by the clustering algorithm in decreasing order of the sizes of their clusters.

was cropped leaving 30 pixels on each side as shown in figure 3.25.

Inspector results: Figure 3.26 shows the training templates used for inspection and suggested by the clustering algorithm for the thermistor screws. The system achieved a defect detection rate of 0.83 (10 defects found) at 2.0 false alarms per defect as shown in figure 3.27.

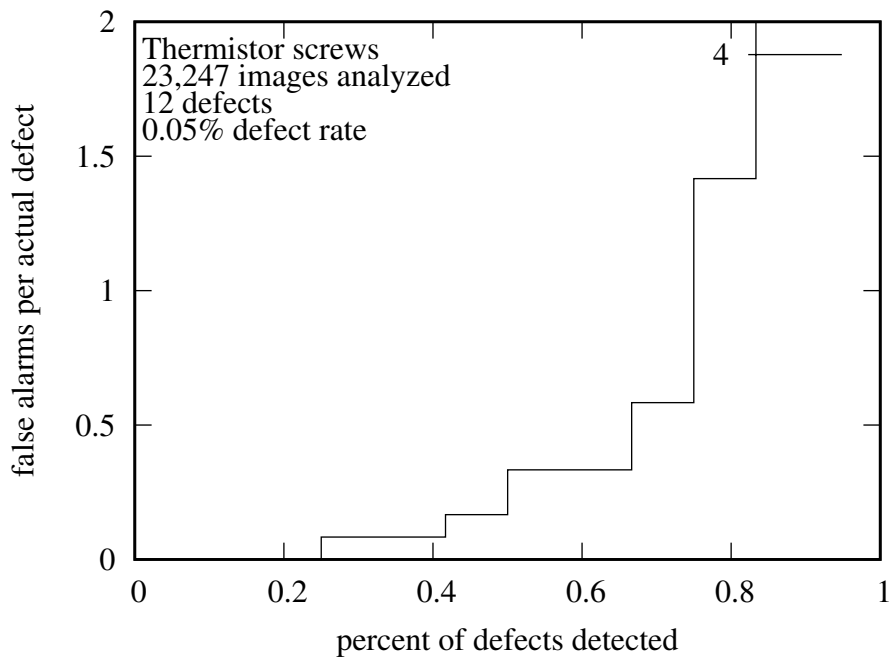


Figure 3.27: ROC for the thermistor screws using 4 training templates. At a 2.0 alarms per defect, the system was able to achieve a 0.83 defect detection rate.



Figure 3.28: The template chosen for the screw near the thermistor.



Figure 3.29: OK seeds of the thermistor screws suggested by the clustering algorithm in decreasing order of their cluster sizes.

3.8 Screw near thermistor

Figure 3.28 shows the inspection template chosen for the screw near the thermistor. This screw did not have much variation in its appearance and it was an easy inspection problem, like the pressure sensor.

Inspector results: The training templates suggested by the clustering algorithm are shown in figure 3.29. The system was able to achieve a defect detection rate of 1.0 with 0.56 false alarms per defect detected as shown in figure 3.30.

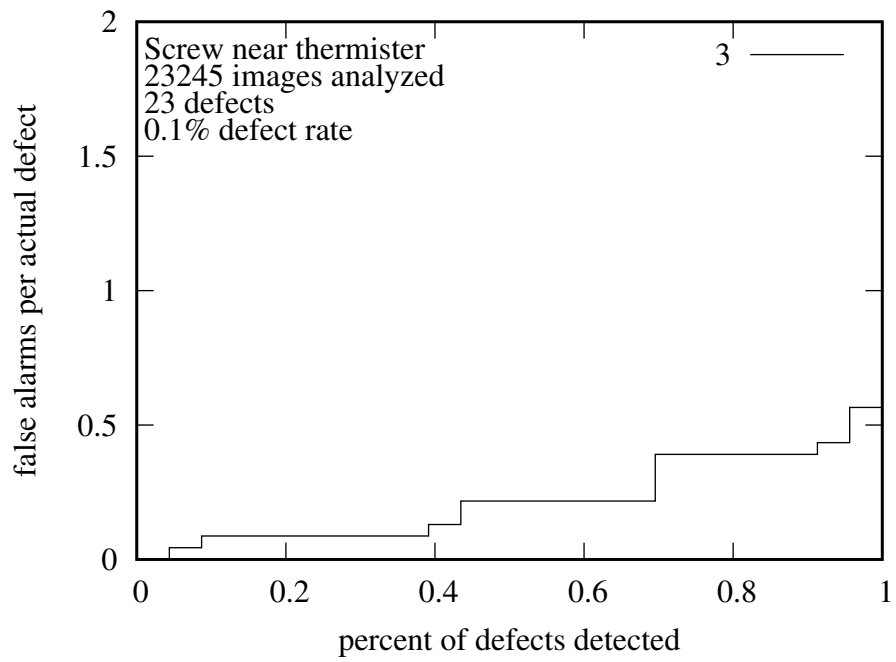


Figure 3.30: ROC for the screw near thermistor using 3 training templates. At the operating point, the defect detection rate is 1.0 and the false alarm rate is 0.56.



Figure 3.31: The template chosen for the screw on the back.

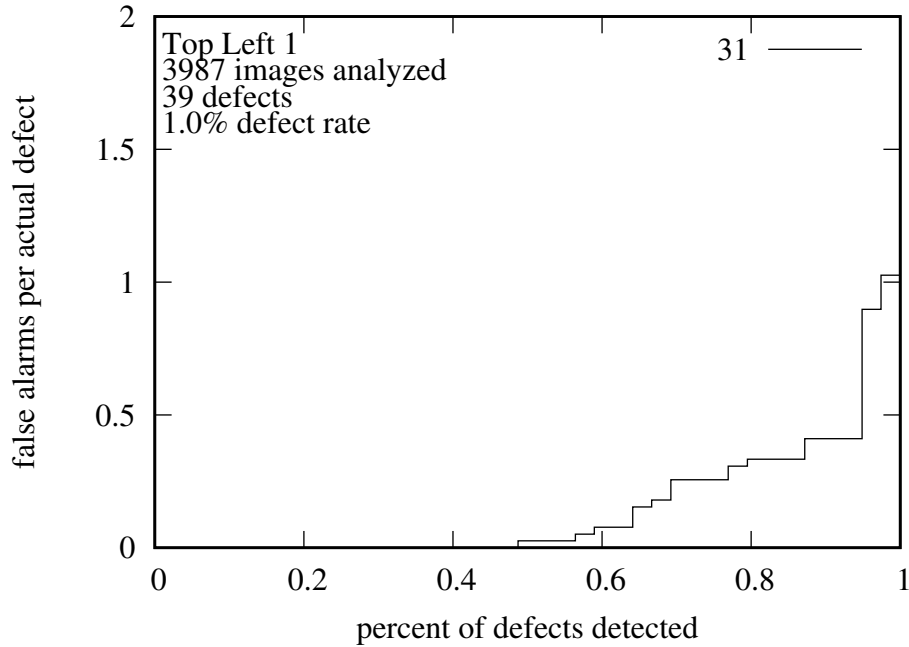


Figure 3.32: ROC of the top left screw at the back of the machine for 31 templates. The system achieves a 1.0 defect detection rate at 1.02 false alarms per defect detected.

3.9 Back screw

Inspector results: As shown in figure 1.15b the screw at the back of the machine has a lot of variation in its color and its background frame color and has different kinds of drives. Even after using edge matching (neglecting color), the system had to be provided 31 training templates (31 largest OK cluster seeds) to capture all the different variations of the screw as shown in figure 3.32, in order to achieve a defect detection rate of 1.0.

Inspection problem	No. of classes	No. of clusters	Defect rate (manual review)	% defects detected	False alarms per defect
White water valve	1	20	0.26 %	100%	0.0
Red water valve	1	20	0.23%	100%	2.68
Tub clamp hose	1	20	0.04%	100%	0.06
EMI filter	2	40	0.08%	97%	2.9
Pressure fensor	1	20	0.02%	100%	0.0
Wire holder	1	60	0.49%	50%	5.0
Thermistor screws	1	20	0.05%	83%	1.4
Screw near thermistor	1	20	0.1%	100%	0.56
Back screw	1	40	1.02%	100%	1.02

Table 3.1: Inspection accuracies for all the uses cases post training.

3.10 Training summary

Our inspection system prototype, post training, achieved a range of 83% - 100% defect detection with a range of 0.0-2.9 false alarms per defect for 8 out of the 9 inspection problems. For the wire holder use case the defect detection rate was only 50 % with a false alarm rate of 5.0. The number of clusters that the clustering algorithm ran for was between 20 to 60 clusters. Except for one (EMI filter), all the other use cases were trained as 1 class inspection problems.

Chapter 4

Conclusion

Our idea of using clustering to find the variability in the appearance of machine parts [7], and training an inspection system for the same, worked well for 8 out of the 9 use cases.

We performed a set of experiments, trying different ways in which we could use the output generated by the clustering algorithm to train an inspection system for appearance variability. Variations included how long to run the clustering, how to select inspection templates from its output, whether or not to use NOK templates in addition to OK templates (1 class/2 class), and when and how to tweak the inspection settings (image pre-processing, IW size, IT size, match method). We were able to train our inspection system prototype for all the inspection problems using one or the other form of these variations, except one, the wire holder.

The machine part named wire holder has a lot of variation in its appearance. The defects for this use case had several forms as shown in figure 4.1. Even after clustering for 60 clusters, our algorithm could only find one of the appearance of the defects as shown in image number 33 in figure 4.2. Also, many seeds were localized at the wrong location inside the inspection windows. Part of the reason is that the



Figure 4.1: Appearance of defects for the wire holder use case. In the first figure, the wire harness is inside the wire holder (OK). Next, the wire holder is not tied and the wire is left dangling outside of the holder (NOK). Next, the holder is not tied and the wire can slip out (NOK). In the fourth figure, the wire is not going through the holder and in the last figure the holder is absent (NOKs).

part does not have a firm structure and occupies very less space inside the inspection window due to its hollow structure, in addition to the variability in its appearance. In this case, one solution to the problem would be to run clustering with those defect templates instead of an OK template or run clustering for an OK and an NOK template parallelly.

Although we haven't completely solved the problem of appearance variability, lessons learned from our experiments were used to build a decision tree to be used as a guide by SEHA engineers working on future inspection problems. We built two

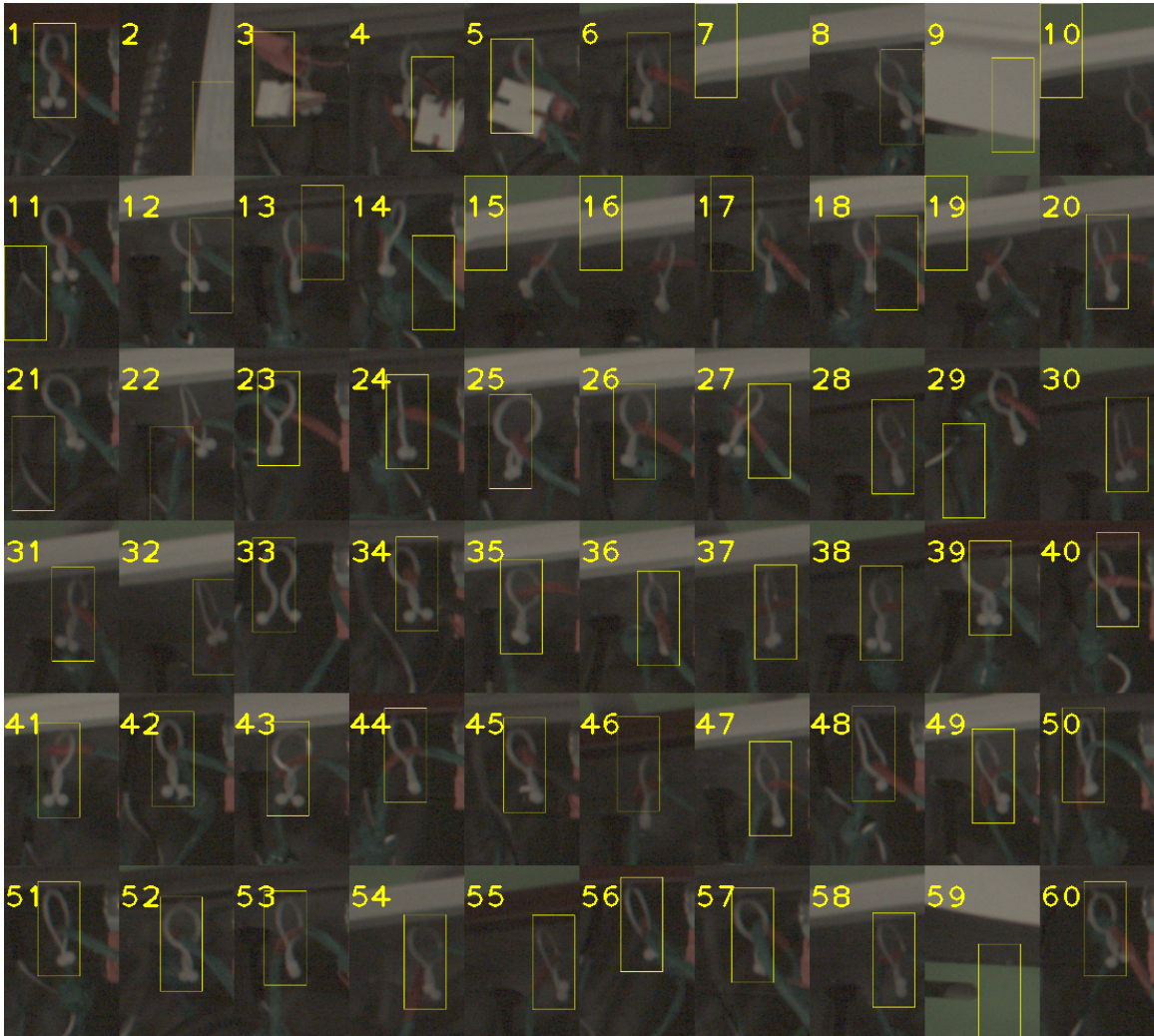


Figure 4.2: The training templates suggested by the clustering algorithm for the wire holder. The only defect captured by the algorithm is image number 33 where the holder is untied and the wire is dangling.

Property	RGB matching	Edge matching
Distinctiveness of the edges of the area of inspection with respect to nearby context	No	Yes
Distinctiveness of the color of the area of inspection with respect to nearby context	Yes	No
Color variation in and around the area of inspection	No	Yes
Occlusions in the area of inspection (wires)	Yes	No

Table 4.1: Guide to choose RGB matching or Edge matching to find the location of the area of inspection.

decision trees, one to tweak the inspection settings and the other to make sense of the clustering output and to train and evaluate the inspection system. Both can work hand in hand to train an inspection system for part appearance variability as shown in figure 4.3, figure 4.4 and table 4.1.

The next step would be to build a software interface (GUI) that encodes this knowledge and enables these options to be intelligently selected for new inspection problems.

Moreover, there is lot of scope for more work. The clustering algorithm could be made faster and more efficient. Instead of starting from scratch (1 template) each time, the algorithm could be changed to start clustering with multiple templates that

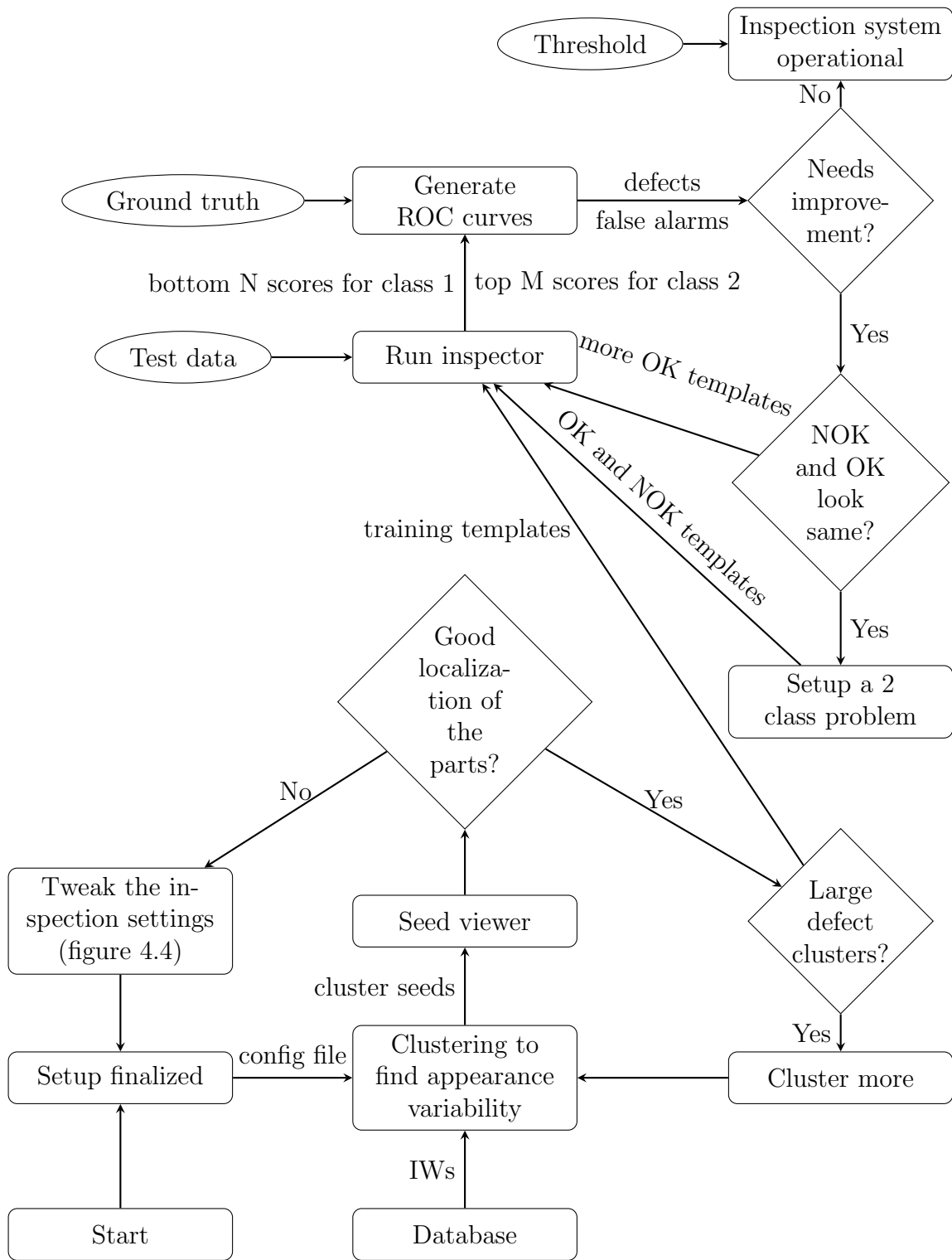


Figure 4.3: Decision tree to help the human engineer train and evaluate the inspection system for appearance variability.

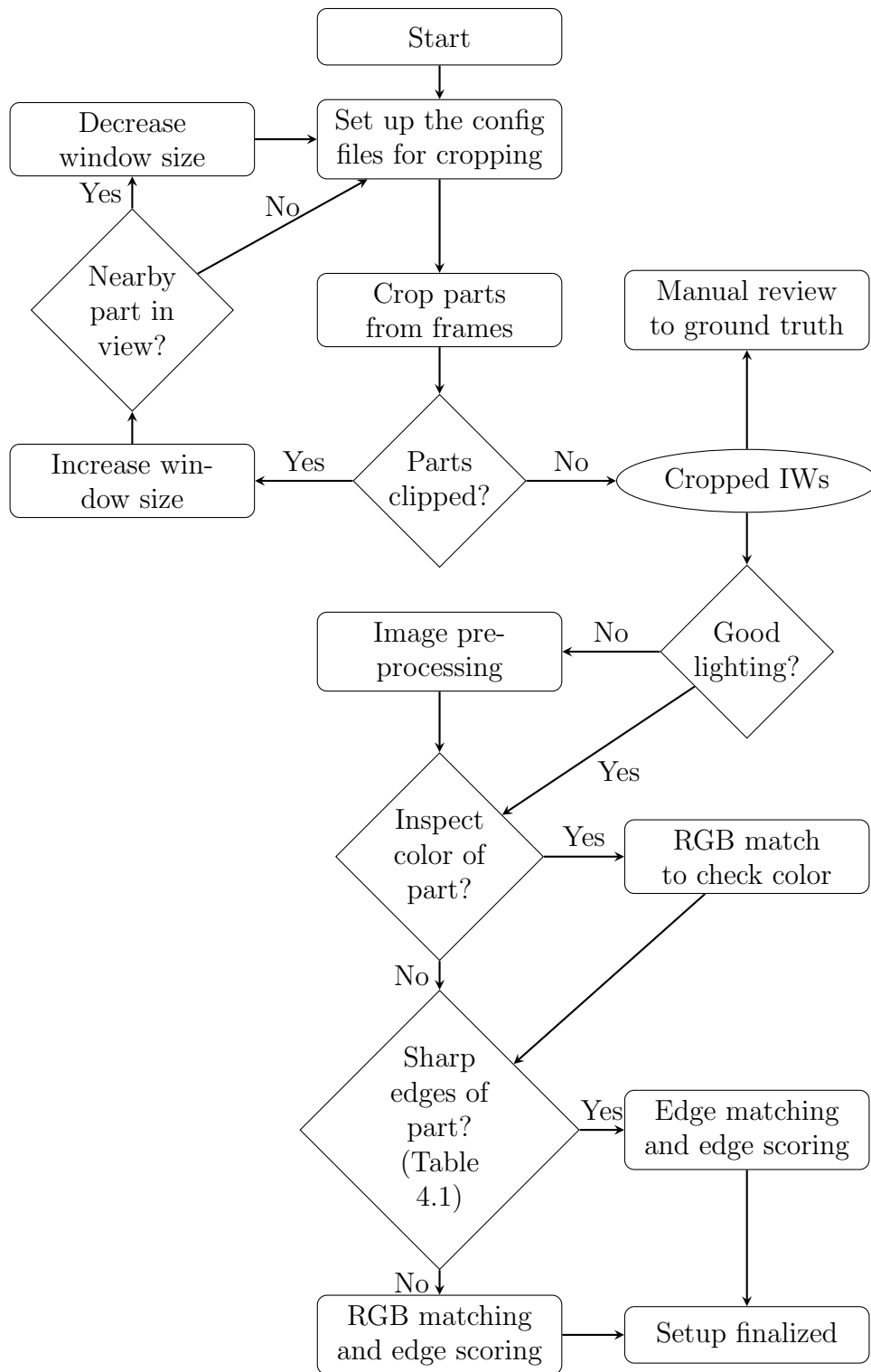


Figure 4.4: Decision tree to help the human engineer tweak the inspection settings.

were already learned in a previous run of the algorithm. We could also have several stages of clustering, where the first stage clusters the images of the part and the subsequent stages cluster specific smaller areas of inspections.

On the training side, research questions like why certain templates match better than others (have large clusters) or why certain defects cannot be detected could be looked into. Our prototype could be incorporated into the current system (Q-eye) or deployed as a stand alone to inspect parts on the assembly line and solve any other potential issues that the system might face.

Appendices

Appendix A Software

Back-end All our code was run on the apollo servers of the college of engineering at Clemson Universities [1]. The code is written in C and uses configuration files and command line arguments as the user interface.

Front-end We did not build a graphical user interface for our system and we ran all our programs on the ubuntu (version 16.04.1) cecas servers. Hence, in order to view and analyze the outputs of our methods on our personal computers, we used an open source implementation of the X window system (Xquartz [12]).

Bibliography

- [1] Apollo servers. <http://cecas.clemson.edu/help/linux/Machines.html>. [Online; accessed 9-July-2020].
- [2] Basler AG. Basler ace acA2500-14gc. <https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca2500-14gc/>. [Online; accessed 9-July-2020].
- [3] Opencv 2.4.9.0. <https://docs.opencv.org/2.4.9/modules/refman.html>. [Online; accessed 9-July-2020].
- [4] Elias N Malamas, Euripides, G.M Petrakis, Michalis Zervakis, Laurent Petit, and Jean-Didier Legat. "A survey on industrial vision systems, applications and tools". pages 171–188. Image and Vision Computing, vol. 21, no.2, 2003.
- [5] W. A. Perkins. "INSPECTOR: A computer vision system that learns to inspect parts". pages 584–592. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. Pami-5, No. 6, 1983.
- [6] S. Baker and D.J Michael. "Semi-Supervised Method for Training Multiple Pattern Recognition and Registration Tool Models". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. Pami-5, No. 6, May 2017.
- [7] Ben Shumpert. "A method to automatically learn part appearance variability in machine parts during appliance manufacturing". <http://cecas.clemson.edu/~ahoover/theses/shumpert-thesis.pdf>, April 2019.
- [8] C. Tsatsoulis and King sun Fu. "Using machine vision for the inspection of industrial assemblies". pages 263–270. NDT International, vol. 19, no.4, 1986.
- [9] Wikipedia contributors. Factory — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Factory&oldid=964353523>, 2020. [Online; accessed 9-July-2020].
- [10] Wikipedia contributors. Image gradient — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Image_gradient&oldid=962141147, 2020. [Online; accessed 9-July-2020].

- [11] Wikipedia contributors. Power over ethernet — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Power_over_Ethernet&oldid=963992676, 2020. [Online; accessed 9-July-2020].
- [12] Wikipedia contributors. Xquartz — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=XQuartz&oldid=965383144>, 2020. [Online; accessed 9-July-2020].
- [13] Barbara Zitova and Jan Flusser. "Image registration methods: a survey". pages 977–1000. Image and Vision Computing, vol. 21, Issue 11, 2003.