Vision System Design: O-Ring Defect Detection

Proposed System

The schematic diagram of the experimental set up used for O seal measurement and inspection is shown in fig. 1. The hardware consists of a light source, an industrial camera, optical lenses, an ultra-white glass platform, a light bracket. A simulation platform was set up to simulate the movement of the production line in industrial environment. The ultra-white glass platform, connected to the rotating platform together, can achieve high precision movement in desired manner under the computer control. According to the paper the model is tested for higher accuracy, so I am sticking with that precision requirement, which has a resolution of 0.01_ and a repeat positioning accuracy of 0.005_. In order to meet requirements of the national testing standard, the measurement accuracy should be higher than 0.08 mm when detecting the O-rings with diameter less than 80 mm.

The isometric view of the setup is shown below, but the constraint is we have single camera which has an automatic timing and movement adjustment with the conveyor when it just completely comes under the area of projection of camera and lens, the conveyor get stopped. But, we need to check whether there are defects on both top and bottom surface of the O-ring, which will be more efficient, for that I am introducing somewhat different concept, that I will be explaining using later figures, and to check the both face we need to flip the ring in such a way that both the faces can be inspected by the camera with the help of using rotatable projection pane having conveyor base (with backlight attached internally with the surface where the ring rests.

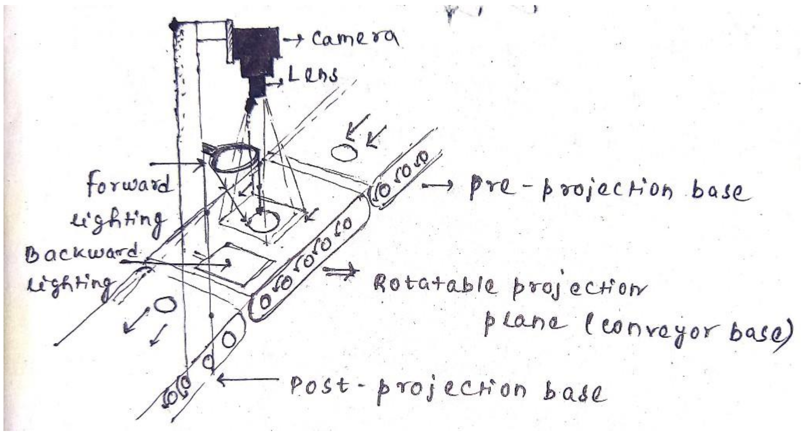


Fig. 1

In fig 1., assuming the individuals ring traverse through pre-projection plane, rotatable projection plane, post-projection plane from one direction to another, when rotatable projection plane conveyor stops on detecting the ring has already under AOP(Area of projection) of the camera lens, the front face with detected. In that situation we are incorporating another rotatable projection plane(In fig. 2) which is upper in level than the plane where O ring rests now(Lower projection plane), and upper plane is connected using two supports with the movable and is rotatable by the rotation mechanism setup.

The upper and lower projection plane both can be rotatable around the axle of rotation ring), also both are movable in to and fro motion from the position of Phase 1 to towards the rotation setup. The rotation ring consists of metallic attachments on one side for(for both projection planes), and that attachments can electro magnetically lock the planes on purpose in an alternate manner.

In the condition of phase 1(fig. 2), front face is being inspected, and conveyor is stopped, after inspection is done, computer trigger the rotational and movement setup so that lower projection plane is pushed away from the beam where camera setup is place and in phase 2, whenever lower plane comes under the upper projection plane in such a way that both the plane somewhat touches the surface,

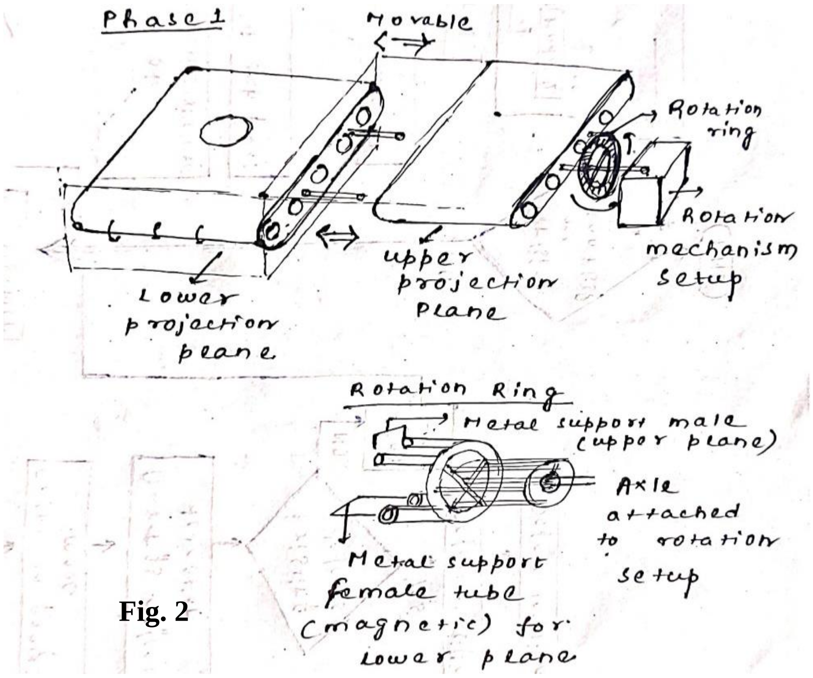

Phase 1
Movable
Rotation ring
Rotation mechanism setup
upper projection Plane
Lower projection plane
Rotation Ring
Metal support male (uppor plane)
Axle attached to rotation setup
Metal support female tube (magnetic) for lower plane

Fig. 2

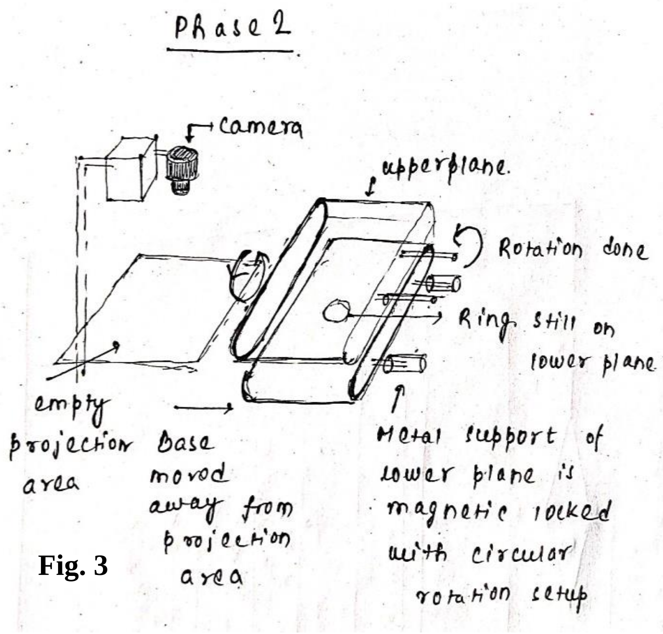

Phase 2
camera
upperplane.
Rotation done
Ring still on lower plane
empty projection area
Base moved away from projection area
Metal support of lower plane is magnetic locked with circular rotation setup

Fig. 3

In phase 2(fig. 3), our task is to flip O-ring by rotating the both planes together, and here up-to now the lower plane is also magnetically locked with the rotation setup and can rotate both plane, and after the rotation the O ring will come on the conveyor base of the upper projection plane(which has backlight attached with the current lower surface as seen in phase 2, which will finally touch the O-ring. After the rotation is done, upper projection plane will come in place of lower projection plane and then the magnetic lock of that plane attached with rotation ring will be released so that it can be moved under the camera AOP for further inspection of lower surface of the O-ring.

Constraints: 1. Rotation will be such that the O-ring will not deformed, and very less movement over the plane of the individual projection plane,

2. If anyhow, the O-ring is displaced the conveyor on the projection plane should auto-adjust the ring so as to bring it just under camera lens setup after the movement.

Camera selection and Setup

The image pickup device of machine vision consists of an array of pixels. In dimension measurement, you can calculate the dimension tolerance based on this number of pixels and the field of view. An important factor for the calculation is pixel resolution, which is the actual length that corresponds to a single pixel of the image pickup device. The pixel resolution can be calculated with this formula:

Pixel resolution = Field of view in the Y direction [mm] / Number of pixels of the image pickup device in the Y direction [pixels] (1)

In dimension inspection, the tolerance used as a threshold for differentiating good and defective workpieces is usually calculated in units of ±5 pixels(fig. 4). This is based on the assumption that the number of pixels that ensures stable tolerance judgment is about 10 times the repeatability. Since the repeatability of typical machine vision is about 0.1 pixel under ideal conditions, the practical repeatability is considered to be 0.5 pixel with some margin included. Multiplying this number by 10 yields ±5 pixels, and this value can be regarded as the minimum unit for tolerance setting. You can use this value to calculate the actual dimension tolerance with this formula:

Actual dimension tolerance [mm] = (Pixel resolution (field of view in the Y direction) [mm] / Number of pixels of CCD in the Y direction) x 5 pixels (2)

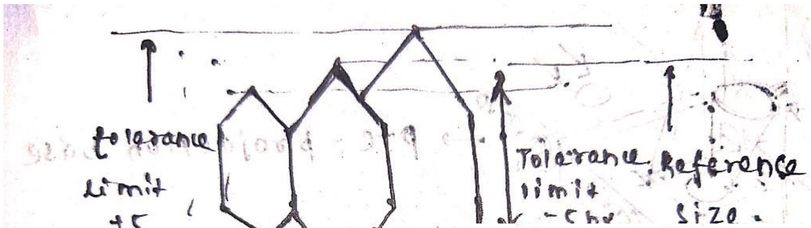


Fig. 4

Camera Setup

The camera setup described in fig. 5(Referenced from the paper, as it is already tested). The forward lighting highlighted the surface defects while eliminating the shadow. Back lighting highlighted the edges, and at the same time can eliminate partial reflection of the forward lighting in the ultra-white glass platform. The specific installation form is shown in, also note that, the surface refers to the projection plane which is clean and transparent, no further scratches, texture, so that backlight attached inside that surface can clearly do work.

According to the requirement of O-rings inspection, a suitable software was modular designed. The specific setup, installations of O-rings quality inspection, dimension measurement. It consists of two separate modules: the size measurement module and defect detection module. Two modules work together to determine whether O-rings are qualified.

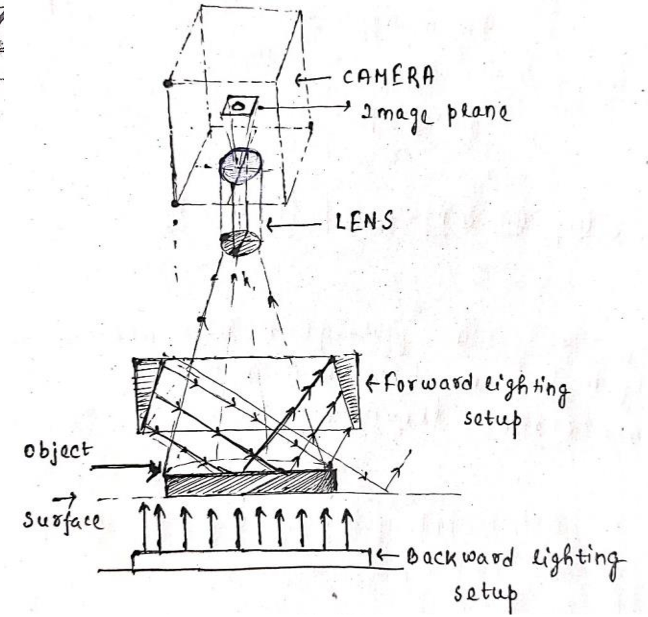


Fig. 5

Proposed Workflow

The workflow designed will be explained based on following flowchart diagram(fig. 6). Our solution model consists of two modules, **1. Defect detection Module, 2. Dimension/Thickness measurement module**. We start with **Defect Detection module,** if defect is detected then flag becomes = 1(which was initialized = 0 while entered the workflow) and the O-ring is sent to rejection queue.

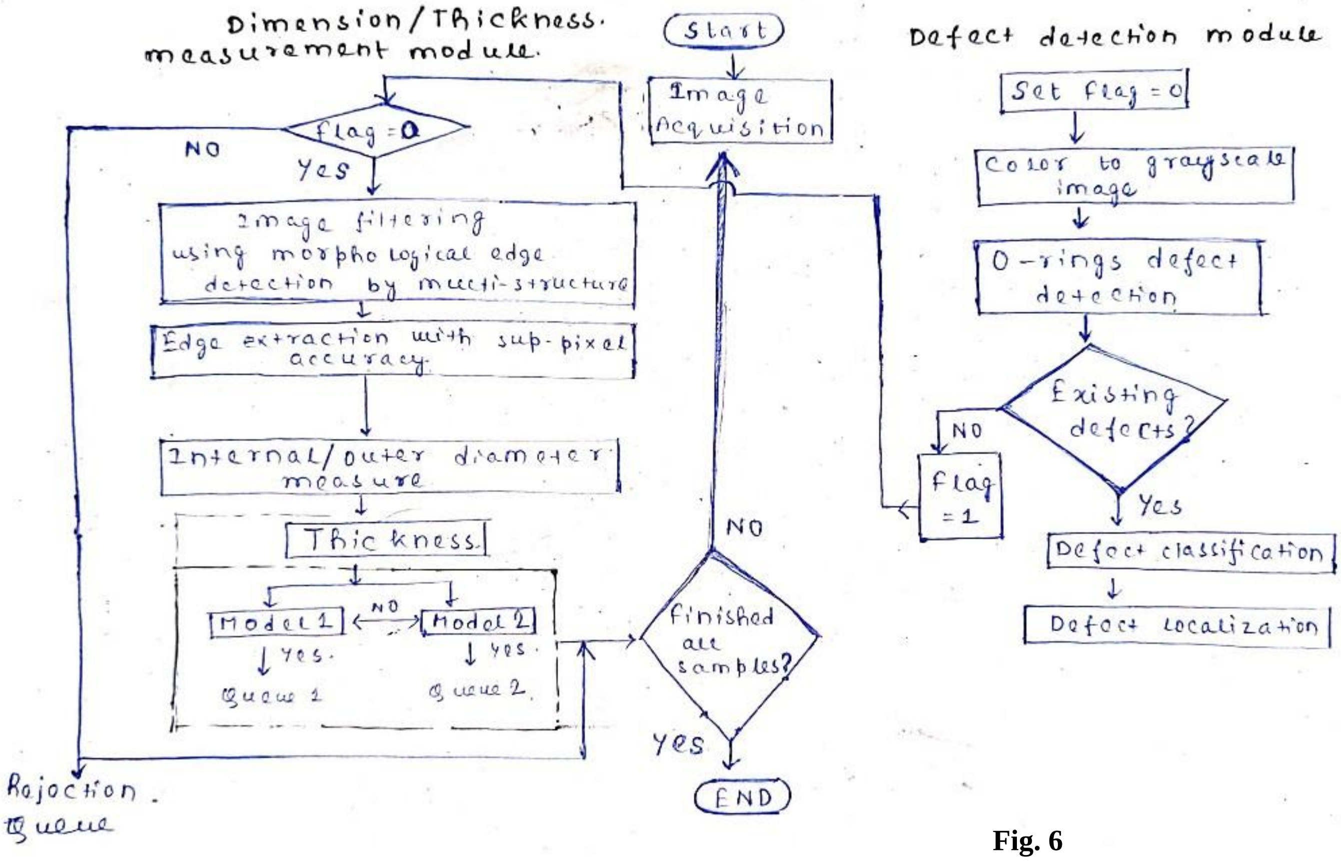


Fig. 6

Dimension/Thickness measurement module

From the workflow we have found first step is image acquisition from the setup mentioned below, then it processed through the defect detection module, if flag = 0, then it comes to dimension module, here first step is to implement an efficient edge detection or filtering.

Edge detection with pixel accuracy

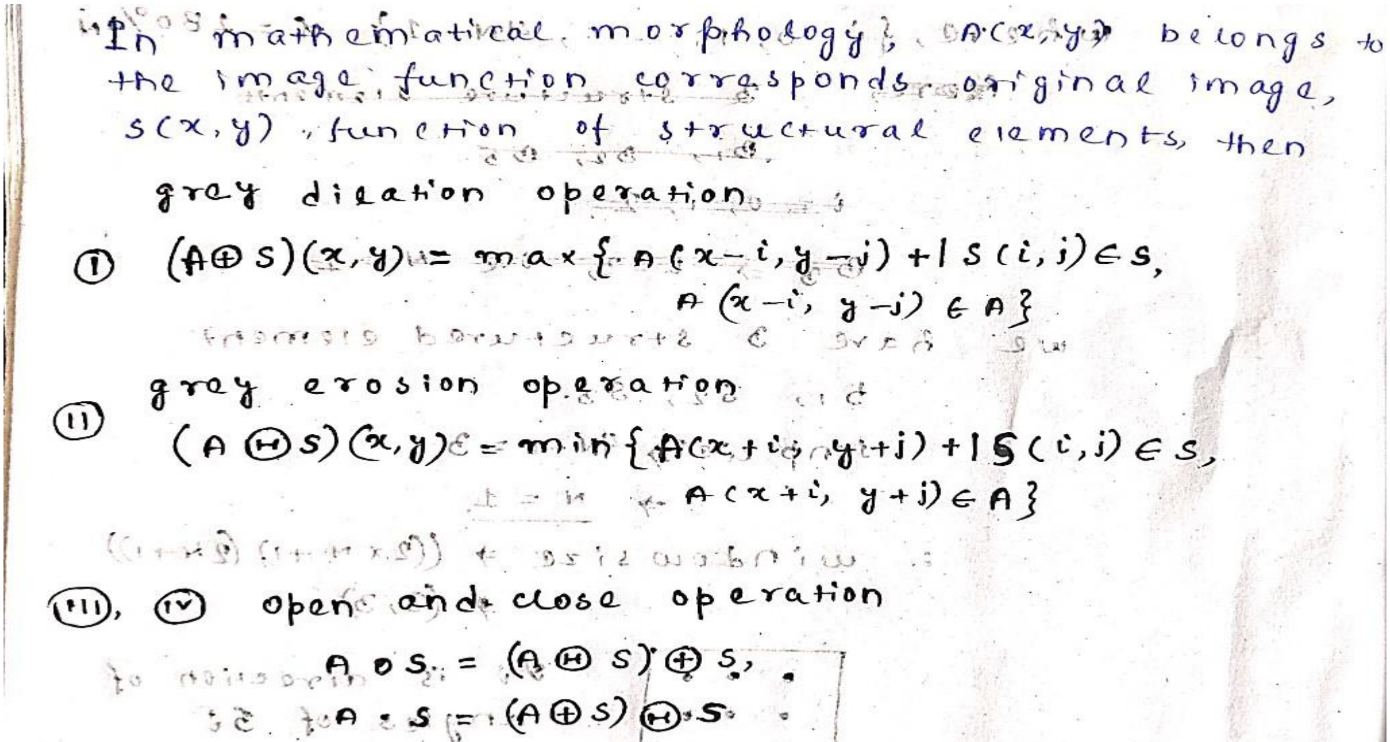
Physical contour is one of the most important properties of an object. In order to extract the contour of O-rings, we must detect its edges. The accuracy of edge detection will directly affect the accuracy of dimensional measurements. People have done a lot of research about this, so a lot of methods were developed to detect edges. One can use simple edge detectors like Roberts, Sobel, Prewitt, LoG, or more sophisticated Canny edge detector. All these methods perform edge detection with a pixel accuracy.

For ring detection, we don't need to extract the seal surface texture, since the edge detection is only used to measure the diameter of the inside and outside. And the extracted texture may affect the measurement accuracy. So we can also use the robust morphological edge detectors, because the morphological methods can eliminate seal surface texture easily.

Basics of morphological edge detectors based on multi-structure elements

Mathematical morphology also called image algebra is the image analysis based on the shapes, where goal is to detect outlines and reduce the unnecessary edges, which is the utmost important here, as we want to ignore the unwanted reflections caused on the surface, surface texture from the O-ring surface, we strictly want to eliminate those to get the exact outline to obtain the shapes.

Morphological edge detection has 4 transformation methods **Dilation, Erosion, Open, Close**. In real life we deal with grey scale data, so some morphological operations are given in fig. 7. The goal of grey dilation here is to maximize A+S, if S is positive, output pixel is brighter, and also output image corresponding to darker region will be enhanced, when dark pixel has lighter pixel around it, so dark unwanted spots get reduced, and for erosion brighter pixel gets darker which helps to remove or eliminate unwanted reflection(region having uniform pixels around a unwanted bright pixel) as it takes min(A-S).



In mathematical morphology, $A(x,y)$ belongs to the image function corresponds original image, $s(x,y)$ function of structural elements, then

grey dilation operation

(I) $(A \oplus S)(x,y) = \max\{A(x-i, y-j) + \mid S(i,j) \in S, A(x-i, y-j) \in A\}$

grey erosion operation

(II) $(A \ominus S)(x,y) = \min\{A(x+i, y+j) + \mid S(i,j) \in S, A(x+i, y+j) \in A\}$

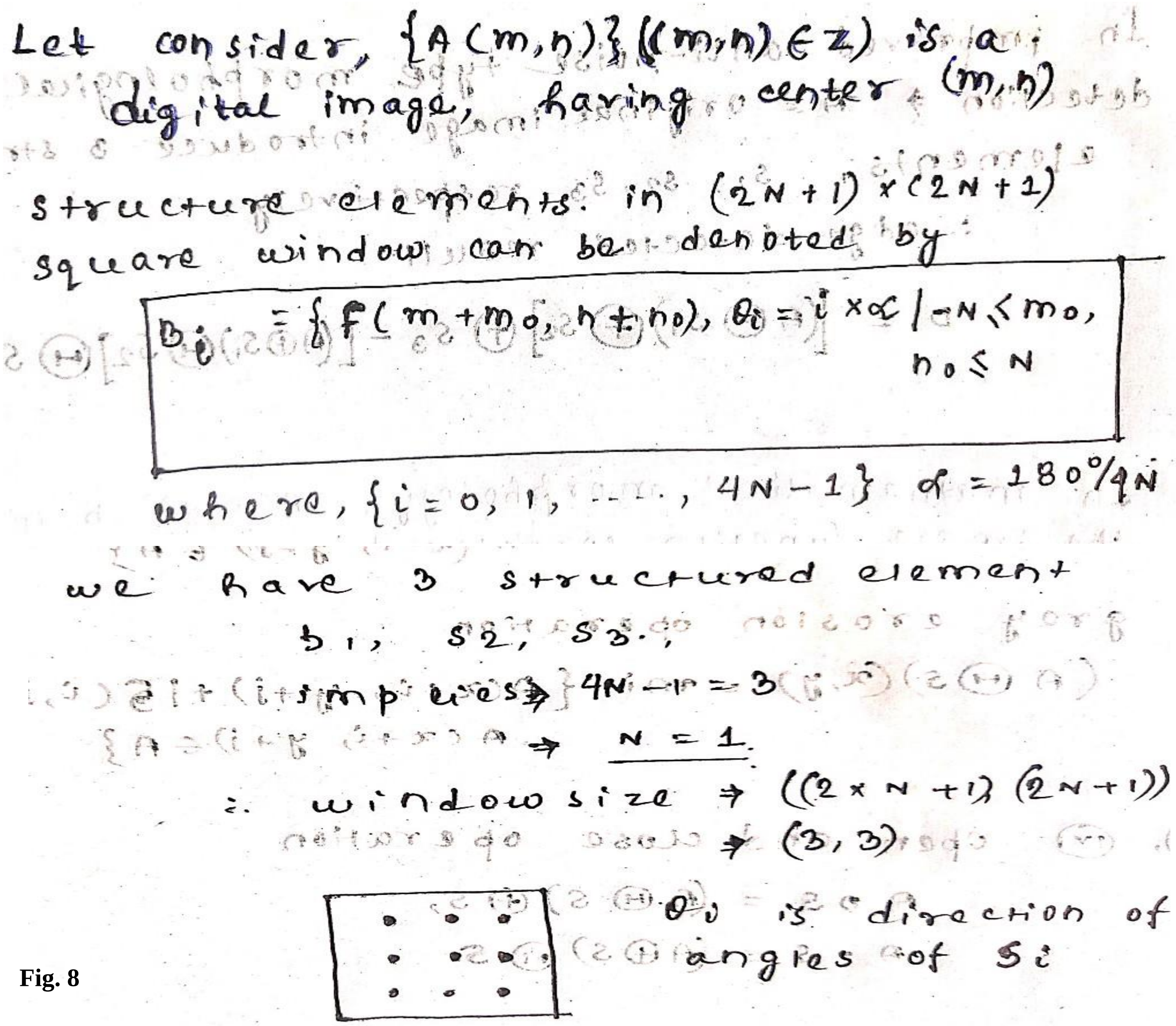
(III), (IV) open and close operation

$A \circ S = (A \ominus S) \oplus S,$

$A \bullet S = (A \oplus S) \ominus S.$

Fig. 7

Now the consideration is about the window size for structured elements, this can be found from following fig. 8.



Let consider, $\{A(m,n)\}((m,n) \in Z)$ is a digital image, having center (m,n)

structure elements in $(2N+1) \times (2N+1)$ square window can be denoted by

$$\boxed{B_i = \{f(m+m_0, n+n_0), \theta_i = i \times \alpha \mid -N \leq m_0, n_0 \leq N}$$

where, $\{i = 0, 1, \ldots, 4N-1\}$ $\alpha = 180°/4N$

we have 3 structured element S_1, S_2, S_3,

it implies $\Rightarrow 4N-1 = 3$

$\Rightarrow \underline{N = 1}$

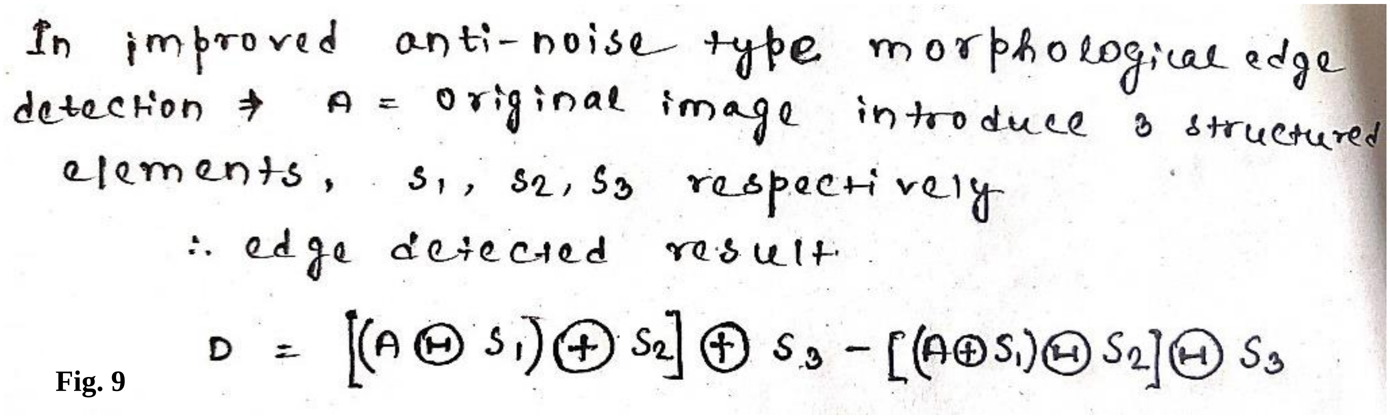
∴ window size $\Rightarrow ((2 \times N+1)(2N+1))$

$\Rightarrow (3, 3)$

θ_i is direction of angles of S_i

Fig. 8

Therefore, an improved anti-noise type morphology edge detection algorithm was used to improve the positioning accuracy by introducing multi-structure elements. In Formula in fig. 9, A denotes the original image, S1, S2, S3 respectively denote three different structural elements.



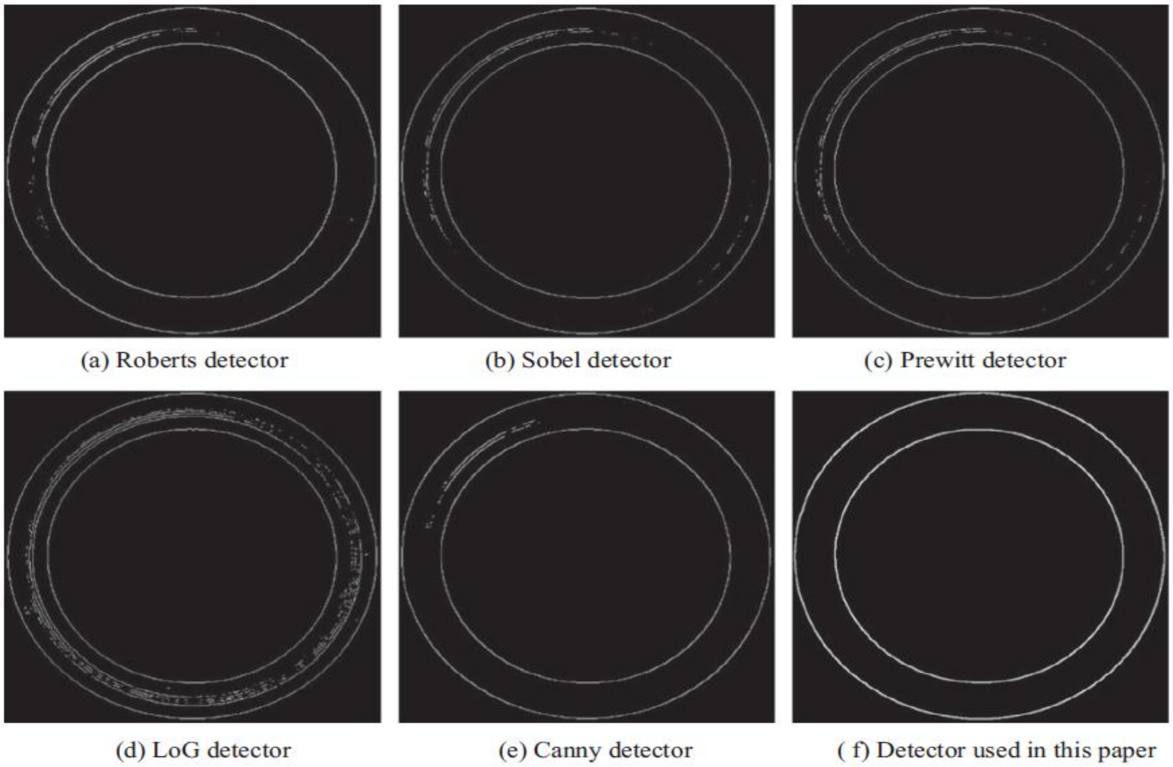
In improved anti-noise type morphological edge detection $\Rightarrow$ A = original image introduce 3 structured elements, S_1, S_2, S_3 respectively

∴ edge detected result

$$D = [(A \ominus S_1) \oplus S_2] \oplus S_3 - [(A \oplus S_1) \ominus S_2] \ominus S_3$$

Fig. 9

fig. 10(Referenced from paper as not tested) shows a comparison of the results by using various algorithms to detect O-rings edge. It can be found that the Roberts, Sobel, Prewitt and LoG edge operator cannot eliminate the unnecessary edges very well. The Canny detector can eliminate some unnecessary edges, but there are still some redundant edges. However, the morphology edge detection algorithm can effectively eliminate the edges of the texture and the reflection-caused edges. So we use morphological edge detection algorithms to extract edges in this machine vision system.



(a) Roberts detector

(b) Sobel detector

(c) Prewitt detector

(d) LoG detector

(e) Canny detector

(f) Detector used in this paper

Fig. 10

Edge detection with sub-pixel accuracy

Sometimes it is necessary to detect edges with sub-pixel precision, especially some industrial applications like measurement of the objects with high precision. In order to further improve the measurement accuracy and reduce hardware costs, we have to detect edges with sub-pixel accuracy. Sub-pixel edge detection is generally determined by pixels around the integer pixel position, as here accurate measure of O-ring concerned in production end, so slight change in pixel location or pixel-level accuracy would cause significant loss in accuracy, so sub-pixel accuracy is important.

Basics of sub-pixel processing

As explained above, the pixel resolution is the actual dimension value of one pixel. In reality, however, dimension measurement using image processing can show dimensions to units less than a pixel through approximate calculation. This is called sub-pixel processing. A sub-pixel is a unit smaller than a pixel.

The sub-pixel processing detects an edge where the contrast changes between bright and dark, converts it into a projection waveform, and then performs differential processing in following fig. 11.

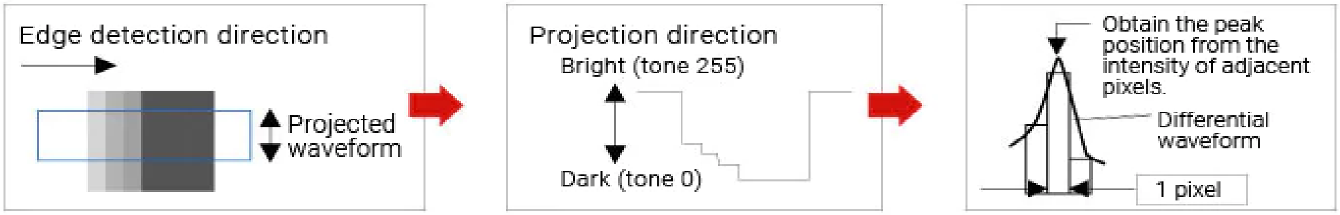
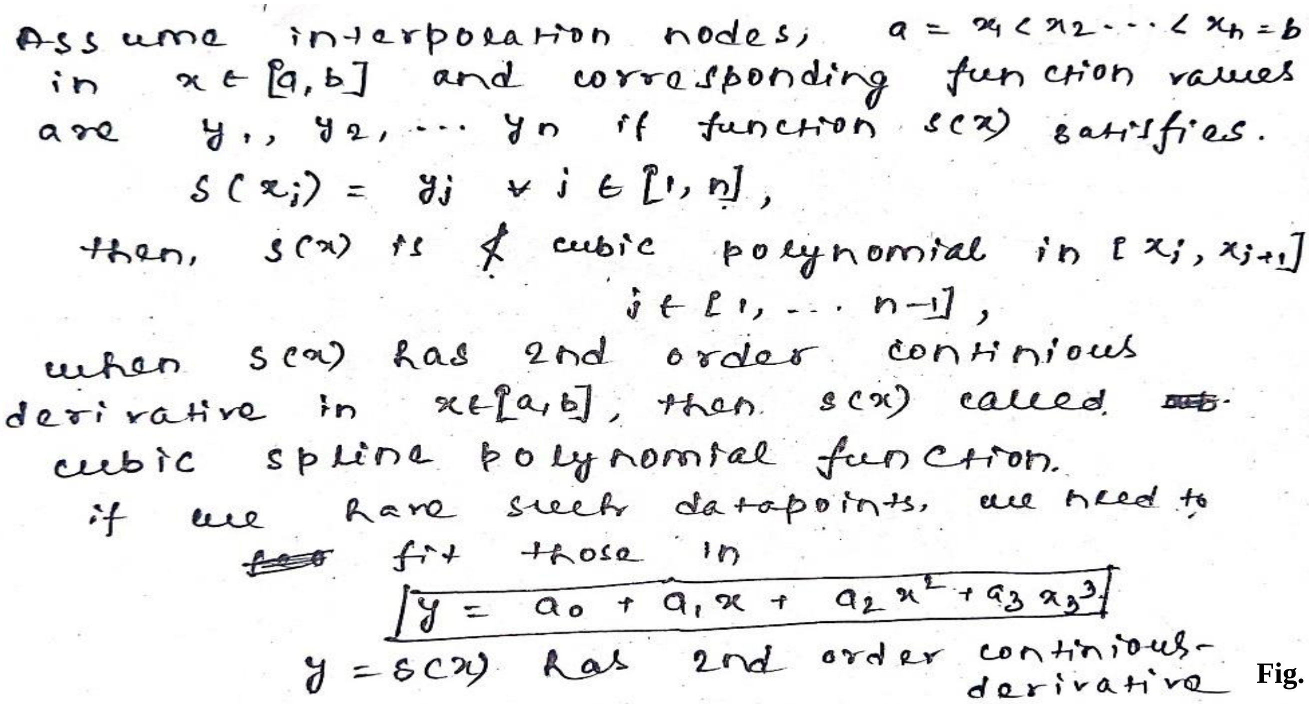


Fig. 11

Most sub-pixel detectors fall in 3 main categories namely fitting, interpolation-based, moment-based methods. After weighing between the requirements on accuracy and computational complexity, a cubic spline interpolation method was selected. This method achieves a sub-pixel accuracy by interpolating the image data to obtain a finer grid of pixels. The ability to obtain accurate interpolation nodes has a direct influence on the accuracy of sub-pixel edge detection. As used herein, the number of nodes is 7(referenced).

Basics of cubic-spline method

Basics of spline interpolation is given below in fig. 12.



Assume interpolation nodes; $a = x_1 < x_2 \cdots < x_n = b$ in $x \in [a, b]$ and corresponding function values are $y_1, y_2, \cdots y_n$ if function $s(x)$ satisfies.

$$S(x_i) = y_i \quad \forall\, i \in [1, n],$$

then, $s(x)$ is ~~a~~ cubic polynomial in $[x_i, x_{i+1}]$

$$i \in [1, \ldots n-1],$$

when $s(x)$ has 2nd order continious derivative in $x \in [a, b]$, then $s(x)$ called cubic spline polynomial function.

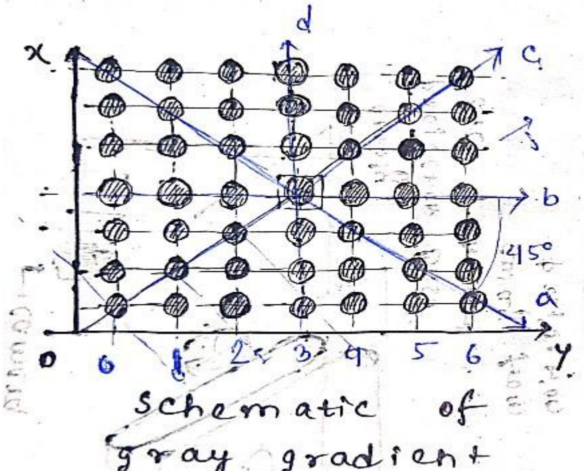
if we have such datapoints, we need to fit those in

$$\boxed{y = a_0 + a_1 x + a_2 x^2 + a_3 x_3^3}$$

$y = s(x)$ has 2nd order continious-derivative

Fig. 12

(referenced) To determine the interpolation node exactly, we need to know the approximate location of the image edge and the gray gradient direction of it. We should find an edge point of edge detection with pixel accuracy, and mapped a rectangular area of 7 x 7 around this point, as shown in fig. 13. Meanwhile, four directions are defined as gray gradient direction in this rectangular area. a, b, c, d denote four gradient directions, respectively. The angle of adjacent direction is 45 degree. And the direction is determined by the absolute value sum of difference between the pixel gray scales, which is calculated as:



Schematic of gray gradient direction

0 to 6 are 7 positions of pixels along line b (gradient direction) where, sum of pixel values

$$E_i = \sum_{k=0}^{6} |P_{k+1} - P_k| \,,$$

$$\underline{(i = a, b, c, d)}$$

Fig. 13

The direction of maximum **Ei** is the gradient direction. And pixels at this direction was extracted as cubic spline interpolation nodes. After obtained the integer pixel coordinates and cubic spline function, coordinates of sub-pixels can be calculated through the zero point at second derivative of interpolating function. **Here, cubic spline interpolation method is found suitable so as to fit the pixel points along the direction of maximum gradient using the cubic spline functions, which fit the points. For cubic function the zero point is the point where gradient changes from one direction to another, to find that double derivative is done, similarly, in case of edge point along the direction of max gradient to find the edge point corresponding to cubic spline, the point at which double derivative of cubic spline is = 0.** The algorithm for this has been attached here.

So this detection algorithm can be divided into five steps:

(1) Get the integer pixels, and a rectangular area of 7 x 7 around it; (2) Find the gradient direction, and pick seven points on that direction as interpolation nodes; (3) Use nodes from step (2) to set up cubic spline function; (4) Calculate the zero point at second derivative of interpolating function to determine sub-pixel points; (5) Repeat the previous four steps until all integer pixel edge points are taken.

(Method 2 referenced) Sub-pixel dimensional measurement algorithm based on intensity integration threshold

Step1: Implement sobel edge detector

Step 2: Gradient direction of intensity levels in a point can be calculated as **[theta = arctan(gradient x/ gradient y)].**

Intensity integration algorithm(IIT)

As for one edge pixel f(x, y), we get its 3 × 3 neighbors, and perform convolution using the horizontal and vertical operators mentioned above. If Gx(x, y)>Gy(x, y), this point can be defined as horizontal edge point, we then extract points along the direction where the intensity levels change along, that is the vertical direction. If Gy(x, y)>Gx(x, y), this point can be defined as vertical edge point, we then extract points along the horizontal direction. Along the direction where the intensity levels change, we extract the intensity of nearby points. Set I as the intensity and l as the location. Figures 14(a) and 14(b) are the I − L figures of the simulated continuous edge and the dispersed edge. The area of the gray part is the integration we require.

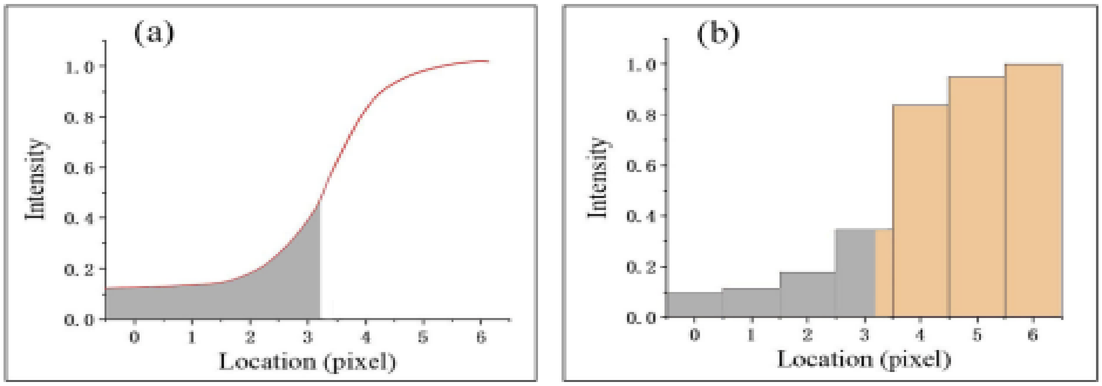


Fig. 14

Then, the intensity integration threshold k needs to be determined. k can be determined by calibration experiments. After determining k, the pixel intensity values are integrated from left to right. When the integrated value reaches the threshold, **here the position t is the edge position of the sub-pixel to be measured as definite integral from o to window length t over location of sub-pixel point l, where the sum crosses the threshold**.

In the actual image, although the pixel intensity can be fitted as a continuous curve [as shown in the Fig. 14(a)], considering the efficiency, we still use discrete intensity values for integration [as shown in the Fig. 14(b)]. In the calculating process, we simplify the integration to summation. At first, arrange the intensity from small to large. If a total of N points are taken, a series of intensity values I1 ~ IN are obtained. Then, calculate the intensity sum Ens of the first n items, which is equivalent to integrating the intensity of the first n points. [**En = sum(Ii) from i= q to n, Find the En,En+1 closest to the threshold**], so that [**$En<k \cap En+1>k$]** The sub-pixel edge position l coordinates where the integral just reaches the threshold. [**$l = (k - En)/n + n$**] The flexibility of IIT algorithm is that the number of points N can be determined according to the blur of the tested edge, and the threshold can be changed according to different lighting conditions to achieve more accurate sub-pixel edge detection results.

Note: Both the above sub pixel accuracy needs to be tested over a standard tolerance limit.

Finally the edge points are found which are like internal and external diameter outline of the O-rings, as shown in the fig. 15.

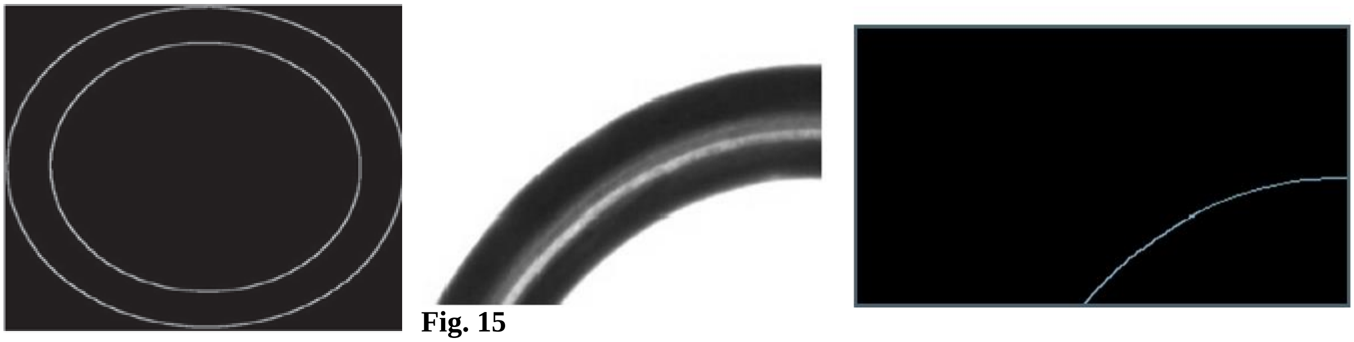


Fig. 15

(Method 1 referenced)Least squares fitting of circles

After sub-pixel edge points are obtained, we need fit them out to measure O-rings diameter and thickness. An improved segmentation algorithm with least squares fitting of circles can be used in order to reduce the measurement error. The original image is divided into eight sections. Each section carry out the least squares fitting method according to the sub-pixel position. Then the center coordinates and the radius of each section will be calculated. The perimeter of whole O-ring is the sum of perimeter of each section. Finally, we can get the inner and outer diameters. Thickness of O-ring can be calculated by the subtracting the inner diameter from the outer diameter. As shown in Fig. 7, the left is one of 8 O-ring sections that need to be extracted, and the right is the corresponding inner edge with integer pixel accuracy. Fig. 8 is the results of extracted sub-pixel edge points. After applying the least squares fitting method for extracted sub-pixel points in Fig. 8, we can obtain the center coordinates Fig. 9 shows the arc lines that were fit by least squares fitting method.

(Method 2 referenced)RANSAC based circle fitting

The details of the RANSAC for circle or arc fitting can be tested for arc fitting through sub-pixel edge points. The details of the RANSAC based curve fitting will be referenced later.

After obtaining the curve fitting for 1st section (45 degree cut for outer/inner circle suppose), we can carry out the fitting for the other 7 sections, and obtain the perimeter of whole O-ring. So the inner diameter was calculated by transferring the pixel units to millimeter with camera pre-calibration. The outer diameter can also be obtained in this way, then **thickness is calculated as the difference between the inner and outer diameter. As we need great accuracy and finer tolerance limit, we can use mean square error so that we can take further step to penalize the errors.**

Fault Detection Module(Referenced and modification suggested)

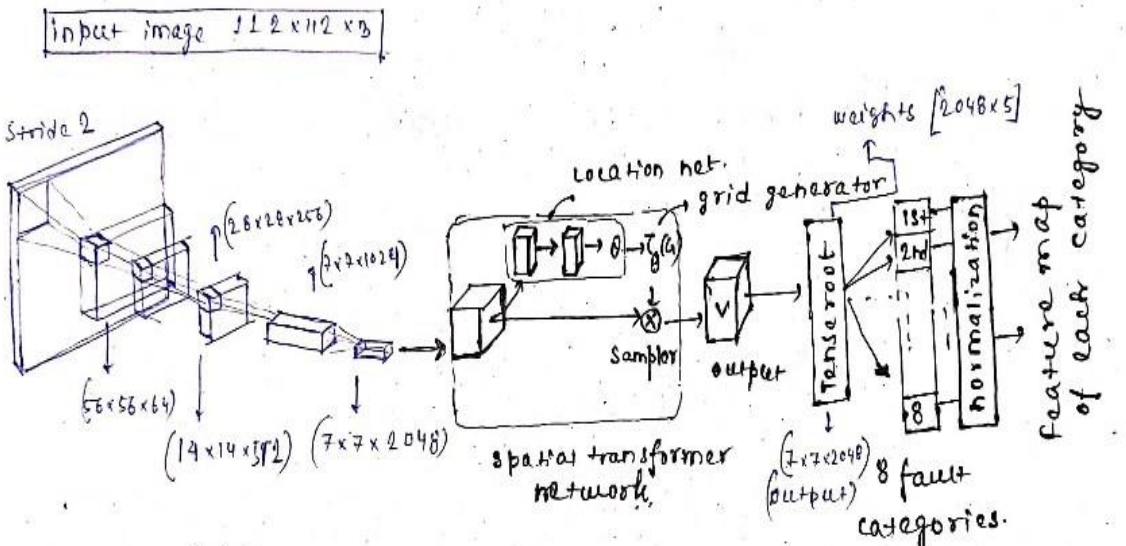
We have to detect 8 different types of faults 1. Excess flash/parting line projection, 2. Flow marks, 3. Foreign material, 4. Mold deposit, 5. Non-fill or void, 6. Off-register and mismatch, 7. Backrind

Implementing deep learning need labelled dataset but it can return great accuracy so we can incorporate attention based deep CNN models.

The top view defect detection application is improved by adding images and type labels to provide a labeling framework for potential machine learning applications. Therefore, quantitative information for more qualitative surface defects, namely inner ring notches and residual glue defects were provided. To detect the surface type defects of the silicone rubber gaskets which have a rough and dim texture, a multi-exposure technique is used to enhance the illumination and highlight the defects. These multi-exposure images are then included in the dataset and trained in a 50-layer ResNet network. The pixel size range of the defects is limited to avoid the feature vanishing during the convolution operations of the network and to allow for more accurate discernment between the background and the defects. Moreover, the deep learning approach can indicate the defect types and defect positions within the predicted image.

In our proposed architecture we tried to implement an attention based model(Spatial transformer network) which helps to better localize or keeps attention to a particular region of interest in the input image. Details regarding the implementation is described in the diagram(The details or architecture of spatial transformer is not specified, as it’s not tested with raw image and labels of 8 different fault classifiers).

Proposed architecture in fig. 16



$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \tau_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

→ output transformed image sampled by sampler, to provide "rectified and resized" version of original input.

Sampling process is bilinear interpolation,

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|)$$

Future work:→ Glimpse network, which replicates retina mechanism.

Fig. 16

Fig. 17 defines the architecture of the network without implementation of attention as referenced in the network.

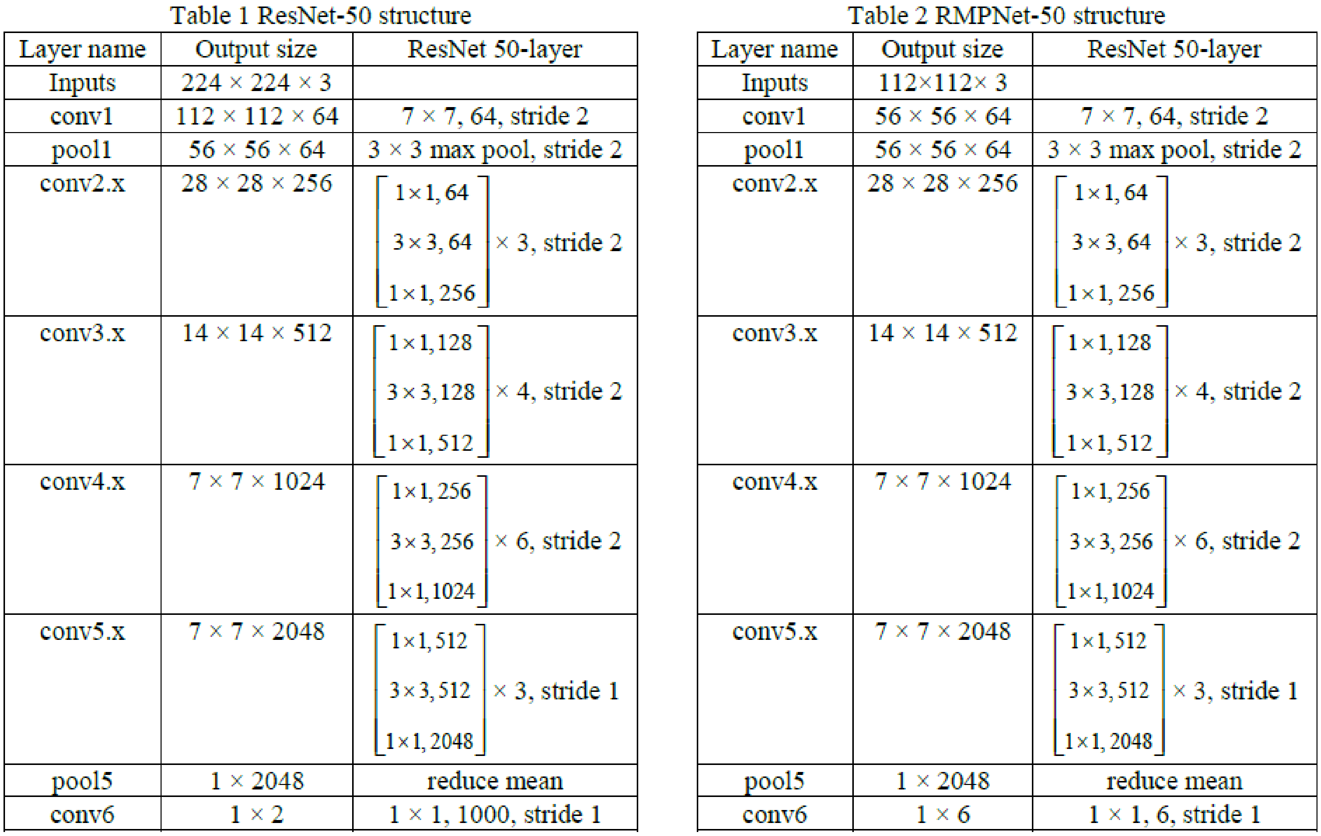


Table 1 ResNet-50 structure

Layer name	Output size	ResNet 50-layer
Inputs	224 × 224 × 3	
conv1	112 × 112 × 64	7 × 7, 64, stride 2
pool1	56 × 56 × 64	3 × 3 max pool, stride 2
conv2.x	28 × 28 × 256	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$, stride 2
conv3.x	14 × 14 × 512	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$, stride 2
conv4.x	7 × 7 × 1024	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$, stride 2
conv5.x	7 × 7 × 2048	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$, stride 1
pool5	1 × 2048	reduce mean
conv6	1 × 2	1 × 1, 1000, stride 1

Table 2 RMPNet-50 structure

Layer name	Output size	ResNet 50-layer
Inputs	112×112× 3	
conv1	56 × 56 × 64	7 × 7, 64, stride 2
pool1	56 × 56 × 64	3 × 3 max pool, stride 2
conv2.x	28 × 28 × 256	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$, stride 2
conv3.x	14 × 14 × 512	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$, stride 2
conv4.x	7 × 7 × 1024	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$, stride 2
conv5.x	7 × 7 × 2048	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$, stride 1
pool5	1 × 2048	reduce mean
conv6	1 × 6	1 × 1, 6, stride 1

Fig. 17

ResNet is chosen as the deep residual network because its residual architecture effectively overcomes the vanishing gradient and explosion issues, and as it consists of residual block we the information of the image is also maintained in the output, which will be better to understand the output with attentioned fault detected region. As the ResNet in TensorFlow pre-training models and related high-level libraries are available in the open-source community, many comparative data for ResNet are available for industrial use. The network proposed in referenced paper uses a multi-class classifier constructed with ResNet 50-layer architecture. Part of the ResNet structure is modified by reducing the size of the receptive field, and the size of the feature map increased as shown in Tables (1)-(2), respectively. The proposed PMPNet network can be seen in Fig. 16. The size of the input picture of our proposed network was 112 × 112 pixels, and there is no max-pooling layer.

Training and dataset for deep learning network(referenced from the paper)

The image of the surface defect and its annotation mask are labeled using the patch-based method. Fig. 9 shows the original image and the labeled defect types for deep learning. This work uses the TensorFlow open-source software library to develop a deep learning network and the corresponding training classifier. In addition to distinguishing the defects, this proposed method can also identify non-defective background images to increase detection accuracy. The training process is divided into two steps. The first step used the pre-trained model of ResNet V1 50-layers for transfer learning. It only trained for five epochs and updated the parameters of the last layer of the convolutional neural network layer. The parameters of the remaining network layers are fixed, the batch size was set to 64, the learning rate is set to 0.001, and the weight decay is set to 0.0004. The second step trained for 50 epochs and updated all network layer parameters, the batch size was set to 64, the learning rate is set to 0.0001, and the weight decay is set to 0.0004. The normalized exponential function is used as the basis of confidence for classifying in the last layer of the network. It is an extension of the Sigmoid function and can be termed as the

$$Softmax_j = \frac{e^{h_j}}{\sum_{j=0}^{C} e^{h_j}}$$

Furthermore, mean square error (MSE) is used to evaluate the loss function. MSE is expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

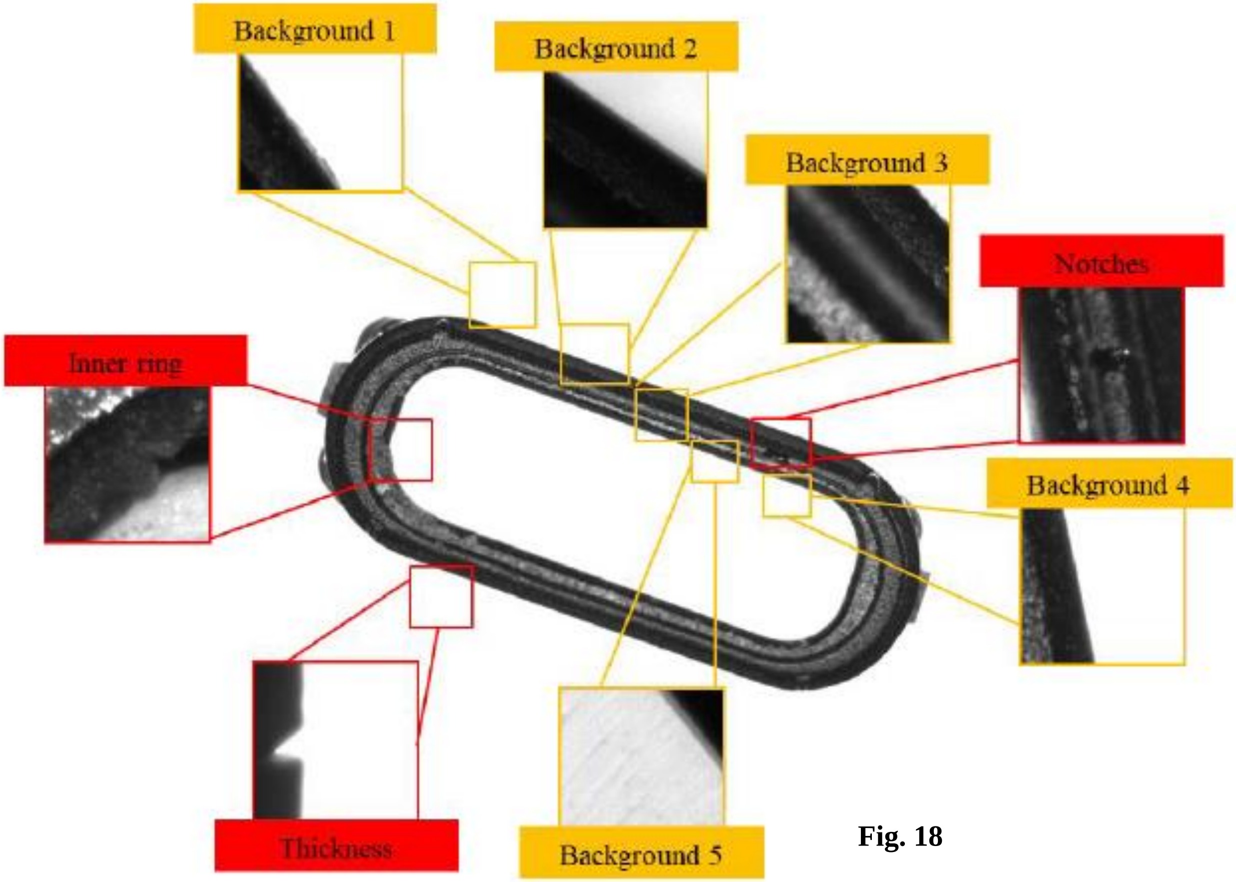


Fig. 18

In this study, a common data augmentation method such as mirroring and up and down flipping is used for image preprocessing.

Defect prediction for deep learning network

The cutting principle of defect detection is the same as the labeling dataset strategy, although the method and parameters were slightly different. The original large image is cut into several 112 × 112 segment patches in the stride of 64 and the trained defect classifier is used to predict the patches. The fused feature map of the defect category is normalized into a gray-scale map. As deep learning method requires a large amount of data to achieve accurate and efficient results. Each silicon rubber gaskets is rotated by a mechanical turntable to acquire more image data from different lighting angles. The feature map corresponding to different colors according to different types of detected defects is shown in Fig. 10. **During the prediction stage, the accuracies of the notch and residual glue defects were 97% and 80%,** respectively. After rotating the objects under inspection at different angles, a series of images are obtained from different angles for the rubber gasket under evaluation. The collected images are analyzed for deep learning and the classification is chosen with voting. **After voting, the prediction accuracies of the notch and residual glue defects were 100% and 97%, respectively. Compare to the traditional rule-based image detection approach, the deep learning approach has high classification performance. It can detect non-contour defects (i.e., notches and residual glue) and achieve valid prediction results.**

Fig. 19 The fused feature map after deep learning prediction

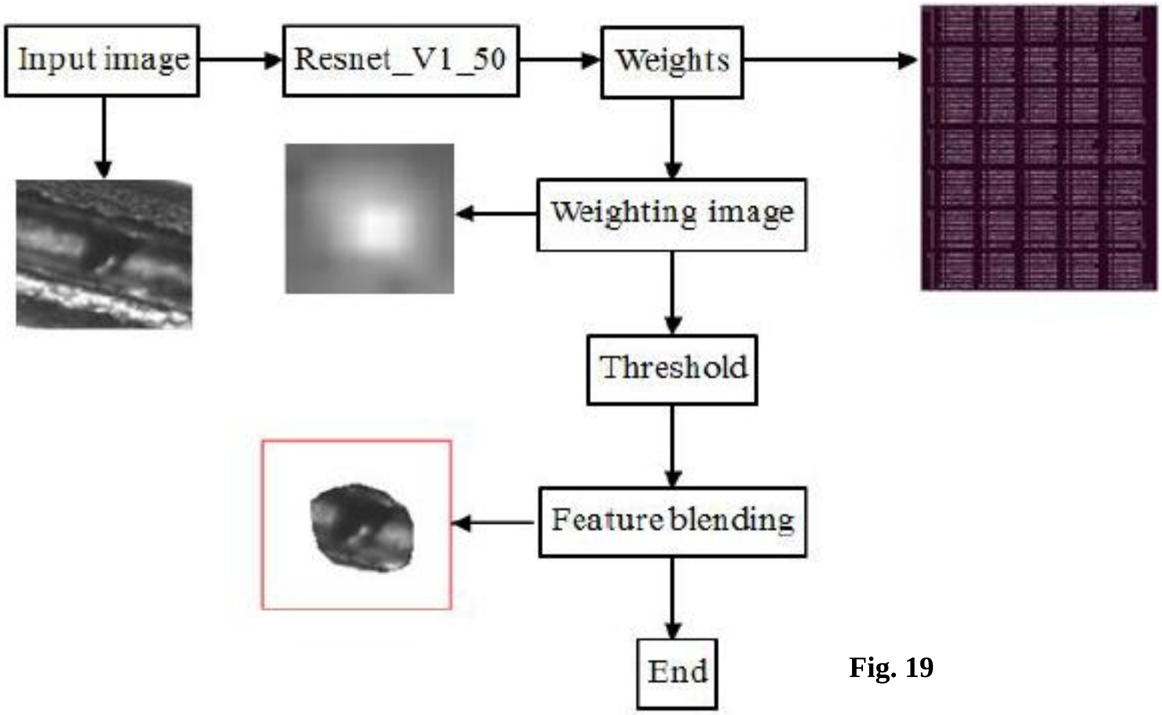


Fig. 19

Reference

In reference I am giving the paper names only

1. Computer vision algorithm for measurement and inspection of O-rings

2. Machine Vision and Deep Learning Based Rubber Gasket Defect Detection

3. Segmentation of Brain Tumor in MRI using Multi-structural Element Morphological Edge Detection

4. Rubber hose surface defect detection system based on machine vision

5. Spline interpolation sub pixel edge detection method based on improved morphological gradient

6. Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges

7. Edge detection method of remote sensing images based on mathematical morphology of multi-structured elements.
8. Research on Defect Detection in Rubber Rings
9. Faithful Least-Squares Fitting of Spheres, Cylinders, Cones and Tori for Reliable Segmentation.
10. Sub-pixel dimensional measurement algorithm based on intensity integration threshold
11. Visual Attention Model in Deep Learning blog.