EDGE DETECTION AND DEFECTS CHECKING OF BINDER CLIP AND WELDED JOINT USING A PYTHON-BASED ALGORITHM: APPLICATIONS IN QUALITY INSPECTION

S. Senthil Murugan[1*], P. Sathiya[2], K. Hariharan[1], J. McJone[1], K. K. Nithiyanantham[3]

[1]Department of Mechanical Engineering, Rajalakshmi Engineering College (Autonomous), Chennai - 602105, India
[2]Department of Production Engineering, National Institute of Technology, Trichy - 620015, India
[3]Department of Aeronautical Engineering, Rajalakshmi Engineering College (Autonomous), Chennai - 602105, India
*Corresponding author's e-mail address: gctsegan@gmail.com

ABSTRACT

Machine vision is a computer vision system that enables a computer to work on image-based inspection and analysis for different applications. In this computer vision, a camera and sensor were used to view an image for its analysis with the help of some sort of algorithms, then processed to infer the image-based data. Machine vision systems along with Python programs can be used for many interdisciplinary applications like weld inspection, online monitoring in manufacturing auto components etc. In this study, the "Edge detection python algorithm" was developed and run through "Google Colab" notebook to inspect the edges and the boundaries of samples like faying surface-modified friction welded dissimilar joints and a binder clip (paper clamp) to check any defects or cracks and straightness etc. With the help of this Python algorithm, the edge detection was done by Sobel, Scharr, and Prewit operators. An input image of the weld joint and the binder clip were converted into Otsu's binary threshold image. The matrix vision camera and the CMOS sensor were used in the machine vision set-up to take the images. This written algorithm is helpful to trace the edges of any kind of solids components. The edges of the binder clips and the weld joint/zone were detected. The binder clips were inspected under two different cases namely the clip in folding condition (Case I) and the binder clip in unfolding condition (Case II). The results showed a defect that was identified in the weld zone and no bending was in the binder clips. This kind of study is useful in manufacturing industries for quality inspection purposes with a new machine vision set up for online inspection of fabricated components like nuts and bolts etc.

Keywords: computer vision, Python, edge detection, algorithm, binder clip, friction welding, Sobel, inspection

1. INTRODUCTION

The computer vision set-up consists of a camera and sensor for image capturing. It gives the machines the sense of sight to allow them to "see" and discover the world through different algorithms like machine and deep learning [1], [2] and artificial intelligence (AI) techniques to offer business equipment the facility to see and investigate tasks in smart manufacturing. This influential technology has quickly initiated the developments in the applications in various industries diagonally, becoming a crucial part of technological developments and digital transformation. Many industries have adopted this technology and embraced a widespread range of automation applications with this computer vision. It assists the industry in increasing production efficiency through automatic quality control to inspect the products in manufacturing sectors, online monitoring systems, reducing human efforts and reducing safety risks [3]. The major application fields of machine vision are defects checkup in objects/jobs [4], edge detection on the fabricated components, barcodes and text interpretation to reveal the products, which is far faster and more precise than manual error-proofing method, assembly and integration, Personal Protective Equipment (PPE) recognition, predictive care, construction management, crop planting and harvest monitoring, automatic weeding systems, facial recognition technology, medical image analysis, self-

Date of submission/acceptance: 31.08.2023/22.12.2023
https://doi.org/10.35219/awet.2023.03

driving cars, intelligent video analytics, Pharmaceutical Drugs [5], and insects finding. Edge detection and geometrical analysis of the object is a kind of image processing technique. Here, the image is extracted and processed. Figure 1 shows a block diagram of image processing techniques to study the data of the images in the machine vision system. Machine vision run by Python algorithm is one of the image processing techniques. Digital Image Processing (DIP) is the use of a computer with an algorithm to process digital images. Unlike traditional image processing, DIP helps us provide a broad range of algorithms to handle issues like noise and distortion during processing. Some of the common computer vision problems include image classification, object localization and detection, and image segmentation. Edge detections are the major one in improving the quality of data used to train the machine learning model. This foundational technique plays a major role in shaping the effectiveness and quality of the machine learning model. With the help of this technique, the object boundaries can be found, and the transitions between different regions encapsulate the structural aspect of an image being provided. Usually, raw images contain noise and fluctuation in pixels, Edge detection aids in noise reduction with relevant intensity or color which simplifies the image with essential information. Precise Object Localization (POL) is the use of a computer algorithm that helps the machine learning model to precisely outline the object contours which predicts the bounding boxes align closely with the actual object of the given image. This precision technique enables the ability to locate the object accurately and recognize an object in different scenes.

The novelty of this paper is to inspect and detect the edges of the faying surface modified friction welded joint and to define the edge/boundaries of a binder clip in different cases (Case I and Case II) with the Python-based edge detection algorithm with Sobel, Scharr, and Prewitt operators. The reason for choosing the dissimilar friction welded joint is to inspect its weld quality and the debonding analysis with the help of a machine vision system as this inspection is not possible visually in industries and the quality of the joint decides the strength of the welded parts. So, this study is needed and it provides the fundamental knowledge on the utilization of machine vision systems and the application of edge detection Python programs to the manufacturing industry [6]. The reason for choosing the binder clip as an object is to inspect its functioning with and without folding its pins and to check for any defects like bending. This study will impart knowledge about the online inspection in binder clip manufacturing industries to check the clips manufactured and to pick the defects-free clip and the binder clips with the edges that are not straight/bent can be rejected by the system. It was difficult to inspect each clip manually as these clips are manufactured under mass production, thus a special automated-computerized system is needed in industries. Installing the edge detection-machine vision system might increase the production of binder clips most efficiently.

The relevance of this work in industries is to make aware of the interdisciplinary application between fields like computer science and mechanical/manufacturing engineering and to produce defects-free products [7] by analysing the various objects with different issues like machining and welding of materials, checking the geometry of the components, detection of defects etc. The input images are converted into different output images including Otsu’s binary threshold image, and the images with the edge/geometry of the joint and clips identified by Sobel, Scharr, and Prewitt operators, Python coding validation and the verification in which image segmentation and edge detection.

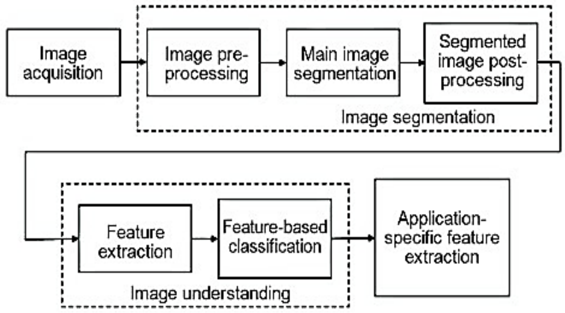

Fig. 1. Block diagram of image processing

2.1. MATERIALS AND METHODS

2.1. Materials

In this study, a binder clip made of spring steel (high yield –strength) [8] and a plastic holder were used for the edge detection. It is a simple temporary fastener for binding sheets of paper together [9].

Further, AISI 304L austenitic stainless steel alloy and AA6063 wrought aluminium alloy were used for fabricating the dissimilar weld joints. The chemical compositions were tested by optical emission spectroscopy (OES) and their results are given in table 1. In AISI304L, the letter ‘L’ stands for low carbon content. It contains carbon (C) of less than 0.03 %. The applications of these materials are plenty in our commercial world. AISI304L is used in industries like food processing, structural/construction applications, aerospace, medical devices, marine applications, metro railway etc. as its excellent properties, especially corrosion resistance [10]. The AA6063 alloy is used in making bicycles, windows and doors, automotive, furniture, aluminium foil, consumer containers, etc. because of its excellent properties and strength-to-weight ratio.

Table 1. AISI304L and AA6063 composition (used for this study-friction welding)

AISI 304L Alloy									
Elements	**C**	**Si**	**Mn**	**P**	**S**	**Cr**	**Ni**	**Fe**	
Weight [%]	0.023	0.38	1.43	0.034	0.009	19.15	8.09	Balance	
AA6063 Alloy									
Elements	**Si**	**Mn**	**Cu**	**Fe**	**Zn**	**Mg**	**Ti**	**Cr**	**Al**
Weight [%]	0.50	0.044	0.029	0.26	0.061	0.41	0.020	0.009	Balance

2.2. Edge Detection and System Design

Edge detection is a paramount first step in many computer vision applications. It pointedly reduces the amount of data required further, it sieved out undesirable or insignificant information. The discontinuity in the image illumination is called an edge. Edge detection is the technique used to recognize the regions in the image where the brightness of the image varies sharply. This sharp change in the intensity value is observed at the local minima or local maxima in the image histogram. The principle behind this is as components/objects pass through the manufacturing plant, the machine vision system takes several scans of the images from different angles to produce a 3D model. When these images are combined, the system allows to identification of the edges and gives significant information about an image [11]. This information is used in the image dispensation to detect the object. Edge detection involves mathematical methods to find points of an image in which the brightness of pixel intensities changes distinctly. The first thing is to find the gradient of the grayscale image, which finds the edge-like region in the 'x' & 'y 'direction. In computer vision, the transitioning of the image from black to white is considered a positive slope which is the measure of how sensitive dependence exists between two variables x and y, whereas a transition from white to black is a negative slope and vice versa of a positive slope. There are a few issues which is similar to the false edge detection, kind of mistakes due to noise, missing low contrast boundaries etc.

Different types of edge detectors [12] are available i.e. gradient-based detectors like Sobel, Prewitt, Robert, and Gaussian-based detectors like Laplacian of Gaussian (LoG), and Canny and modified declivity operator (MDO). It has been shown that the MDO gives a better result compared with other edge detectors because it detects low amplitude declivities that are edges and combines weak edges with strong edges to identify the images removing noises [13]. MDO is a type of edge detector similar to the computer vision edge detection technique, it is a specialized operator. It is used to identify edges based on declivity like a slope or a gradient in an image. Since it is a hypothetical operator there is no such operator establishment in that name. If such an operator exists it would be more advantageous in traditional edge detection techniques.

In this study, the input images were processed by Sobel, Scharr, and Prewitt operators through the Python program and their output is compared.

Edge detection could be detected by image acquisition and processing [14]. Figure 2 shows the schematic diagram of the Edge detection which shows the system design used in this study. The significance and relevance of this specific system design in the context of binder clip edge detection is that binder clips are a common object with varying sizes, shapes and materials which makes them challenging for accurate edge detection. The machine system mainly consists of a camera (matrix vision), and a sensor (CMOS) to capture the image. The captured image was now ready to process through the Python program.

In this study, Python coding was used for the image processing. The Python program can be assessed through the "Google Colab" notebook, which helps run the Python. The reason behind using Python as a programming language and Google Colab as the platform is because Python is simple and has a clear syntax which helps users in a more particular way and the application of Python is emerging in the mechanical engineering field. Since Python is more integrated with machine learning its libraries and frameworks are well-suited for image processing and edge detection techniques [15]. "Google Colab" is a notebook provided by Google. Google Colab provide a free platform for running Python codes this is particularly used for educational and research purpose. It helps in free cloud computing, GPU acceleration, pre-installed libraries, easy sharing and collaboration. The pre-installed libraries which are mentioned above in Python make it convenient to import and use this library for various tasks like image processing and edge detection. Google Colab's GPU acceleration significantly speeds up the execution of edge detection algorithms. The captured images by the camera of the weld joint and binder clips are retrieved (image acquisition) from the system through the edge detector (Sobel, Scharr, and Prewitt operators) for further image processing. The edge detectors remove the noise and the background available on the images. Then it creates threshold segmentation with Otsu's binary threshold image and then the boundaries/edges of the object are identified with some output images. Thus, edge detection helps in real-world applications like quality control in various sectors and manufacturing parts. Binder clips have irregular edges, posing challenges to edge detection systems. The proposed system can improve the precision and accuracy of detecting edges in complex objects like binder clips, welding joints etc. This algorithmic-machine vision innovation can potentially introduce new concepts or improvements that extend beyond the specific domain of welds and binder clips which contribute to computer vision and image analysis.

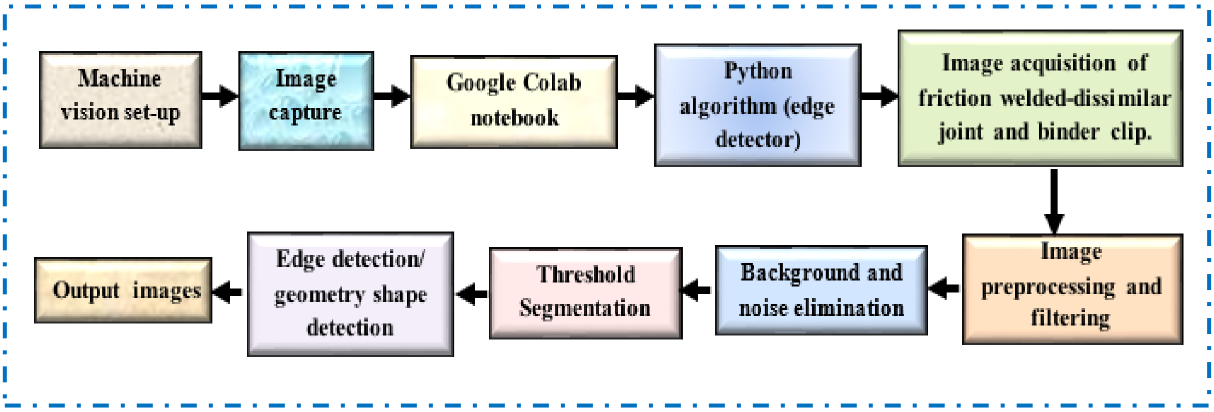

Fig. 2. Schematic of edge detection in this study

2.3. Friction Welding

Friction welding (FW) is a solid-state welding process that joins two materials through the generation of heat from friction between their surfaces [16]. This welding technique is commonly used in various industries for its advantages in producing strong and reliable welds. FW works on the principle (Fig. 3a, 3b) of generating heat through mechanical friction. Two materials to be joined are brought into contact under pressure, and one of them is rotated at a high speed. The friction between the two surfaces generates heat, softening the material at the interface, and eventually, the materials bond together as they cool. In general, the joining of dissimilar alloys is a challenging one as they have different elemental compositions and different material properties.

In this study, two dissimilar cylindrical rods (AA6063, AISI304L) were joined using the FW process (figures 3c, 3d) at 15 bar friction pressure, 24 bar upset pressure, and 3 sec friction time. The faying surface of the AISI304L (SS) specimen was modified as shown in figure 2f using lathe m/c. A weld specimen was of size 101 mm in length and ϕ12 mm in diameter. Specimens (Fig. 3e, 3g) were frictionally welded together. Figures 3h and 3i show the images of the welded dissimilar joints before and after machining respectively. For the edge detection, a sample was cut (a box shown in figure 3i) using a wire EDM machine just 10 mm away from both sides of the weld interface (WI), and then the specimen was polished.

3. RESULTS AND DISCUSSION

3.1. Joint Analysis

After welding, the formation of a sparrow's nest-like flash shape (Fig. 3j) on the friction-welded specimen was noted without any damage. The WI was also well and narrow (Fig. 3i). The axial shortening was around 13.0 mm since the total length of the welded specimen was 189.0 mm. Axial length reduction happened on the aluminium side more than that on the steel side. The Quantum of weld flash was great in the middle of the joint. As far as this method is concerned, the working parameters were adequate to fabricate joints and to cause ring-shaped flash formation.

3.2. Input Images

The pictures of the objects for which the edges/boundaries to be detected were taken with a machine vision camera and given as input. It is important to cleanse the objects before snapping photos. The captured input original images (Figs. 4a to 4c) were stored in the system, and then the address where they were stored was given in the Python program (edge detection). Figures 4a and 4b show the original input images of the binder clips with (Case I) and without folding (Case II) states respectively. Figure 4c shows the original image of the faying surface modified AISI304L/AA6063 dissimilar metals friction weld joint. This joint was made with the help of a continuous drive type rotary friction welding machine at a constant chuck speed of 1500 rpm [18], [19]. The image is marked with which sides are responsible for which materials. The edge detection works quite well but sometimes it may not work properly due to the complex or noisy background which leads to false edges. This can be rectified by fine-tuning and advanced algorithms. The photos taken are to be clear and should have no noises and a background clear. Figure 4d is the sample of the FW joint after the milling process. This defective sample is used here for identifying its edges and also for surface defects detection. The V-shaped WI was found on the sample this is due to the tapering on the AISI304L side.

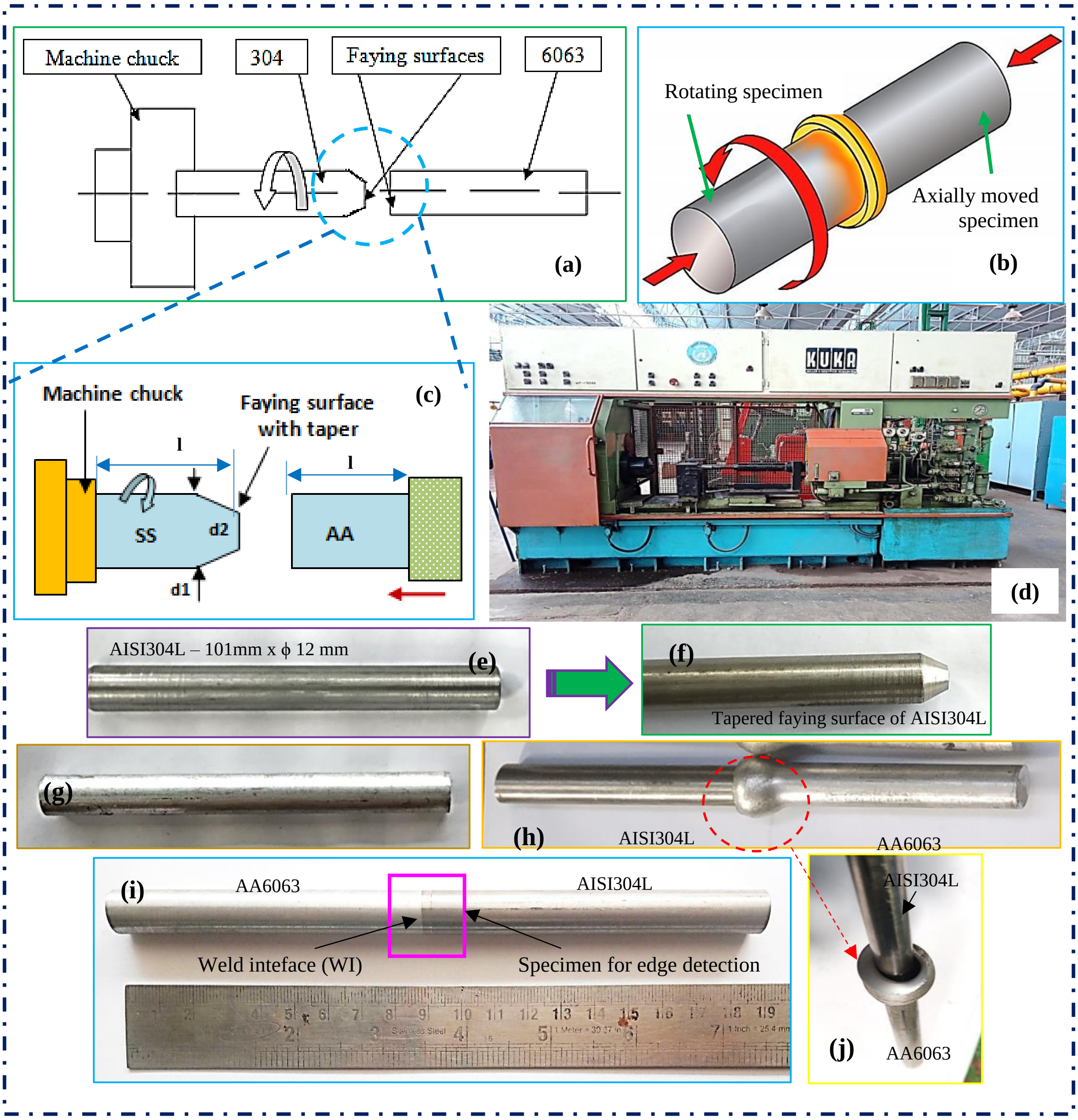

Fig. 3. a) FW concept used in this study [17], b) FW principle, c) Specimen design and FW method for this study (where d1= 12 mm, d2= 6 mm, l=101 mm), d) FW m/c, e) AISI304L specimen, f) faying surface modified AISI304L, g) AA6063 specimen, h) FW as-joint, i) FW welded joint after machining, j) joint like sparrow's nest

3.3. Output Images

3.3.1. Binder Clip Folding State (Case-I)

For case, I, the binder clip of the closed pin condition was taken for edge detection. Figure 4a is the image that was given as the input to the algorithm for the image segmentation. The corresponding Otsu's binary threshold image was obtained as the output result, which is given in figure 5a. Otsu's binary threshold is used to separate objects from the background by the maximum variance between the foreground and background pixel intensities. It gives a clear edge detection from a clear binary image. The image was processed by various edge detectors such as Sobel, Scharr, and Prewitt operators and the output was obtained with the help of a python algorithm. The obtained output images of Sobel, Scharr, and Prewitt operators are shown in figures 5b, 5c and 5d respectively. From the threshold image result, a single steel bar is only showing and does not show its edges. The plastic and the steel are in the same brightness. However, the Sobel output shows the images of binder clips with clear boundaries. The waviness (irregular shape) on the right plastic side of the binder clip and the dust particle sticking on the steel part is detected and shown by all edge detectors. It is hoped that there is no limit on the object's materials as it detects plastic material also. Similarly, the Scharr and Prewitt outputs are shown here. It is worth noting that comparing the Sobel, Scharr and Prewitt outputs, the Prewitt image is far better compared to others as its image is clear, bright and accurate. Though the Sobel detection is accurate, it does not give bright visibility compared to others. The Prewitt detection is recommended for this detection. With the help of this study, the geometrical shape of the binder clip can be identified and its dimensions including thickness can be measured. The sharpness of the edges is somewhat good in Prewitt's output.

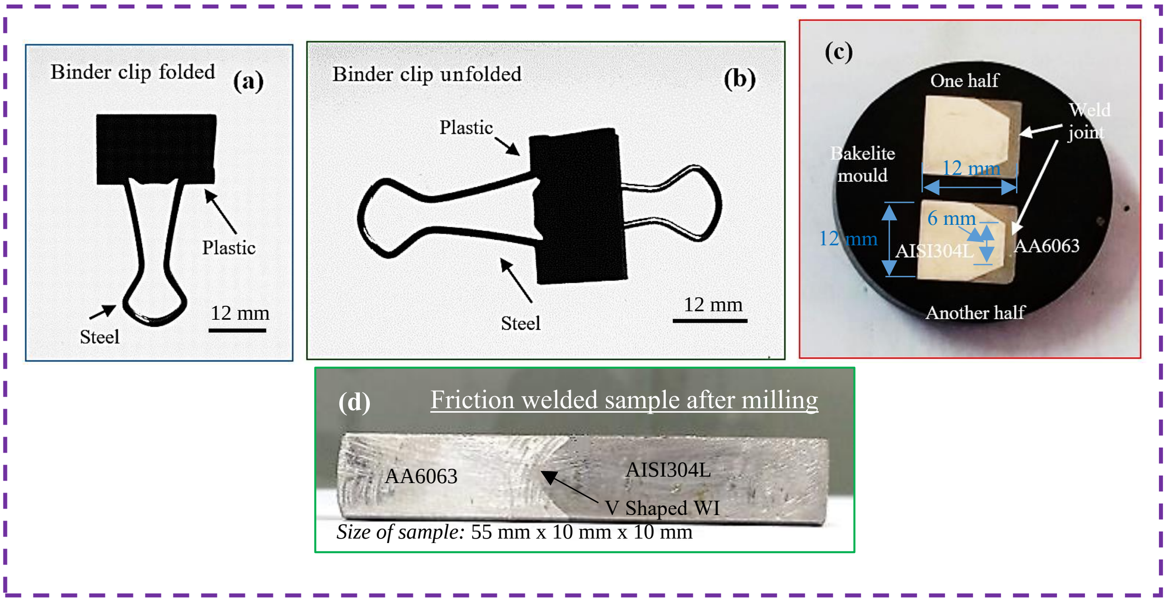

Fig. 4. Input original images of samples. a) Binder clip folding case, b) Binder clip unfolding case, c) dissimilar friction weld joint, d) dissimilar weld sample for edge and defects detection, image taken by mobile camera.

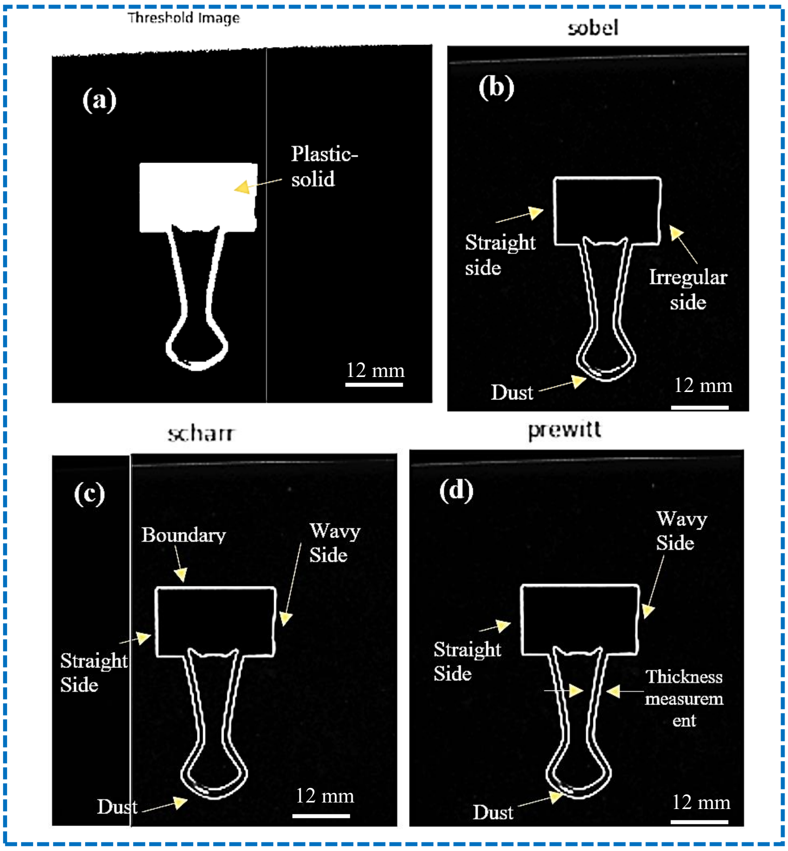

Fig. 5. Folding state. a) Otsu's binary threshold image, b) Sobel output c), Scharr output (d), Prewitt output

3.3.2. Binder Clip Unfolding State (Case-II)

In case II, the binder clip pins were stretched open (unfolding state) like the sample in figure 4b. Figure 4.b was given as an original image input to the algorithm for the edge analysis in case II. The algorithm was processed and provided the output results. The obtained output images are given in figures 6a to 6d. The result shows its ability to identify the edge clearly with different images. The corresponding Otsu's binary threshold image is given in figure 6a. The Sobel output, Scharr and Prewitt output images are given here from 6b to 6d respectively. From the images, the Sobel operator performs a little bit dull compared to others and its accuracy is also somewhat less in identifying the very small particles or discontinuities in samples. For example, small dust particles (mark 1) found in figures 6c and 6d are not available in figure 6b (Sobel output), which shows the failure of the Sobel operator in the accuracy test. Since mark 2 (Fig. 6d) is not straight with the object's placed plane, it is to be understood that this object is inclined with the placed plane. The Scharr and Prewitt operators' outputs are almost the same and clear for the binder clip samples. The output has been aligned with the expectations and prior research. The sharpness of the output results (Figs. 6c & 6d) is somewhat good. However, all the operators discussed in this paper are eligible to analyse any complex object which may have irregular size and edges. As a result, edge detection shows the identification of the image boundaries. These have high potation in real-time application in various fields of object detection. The measurement of the binder clip also easily can be identified.

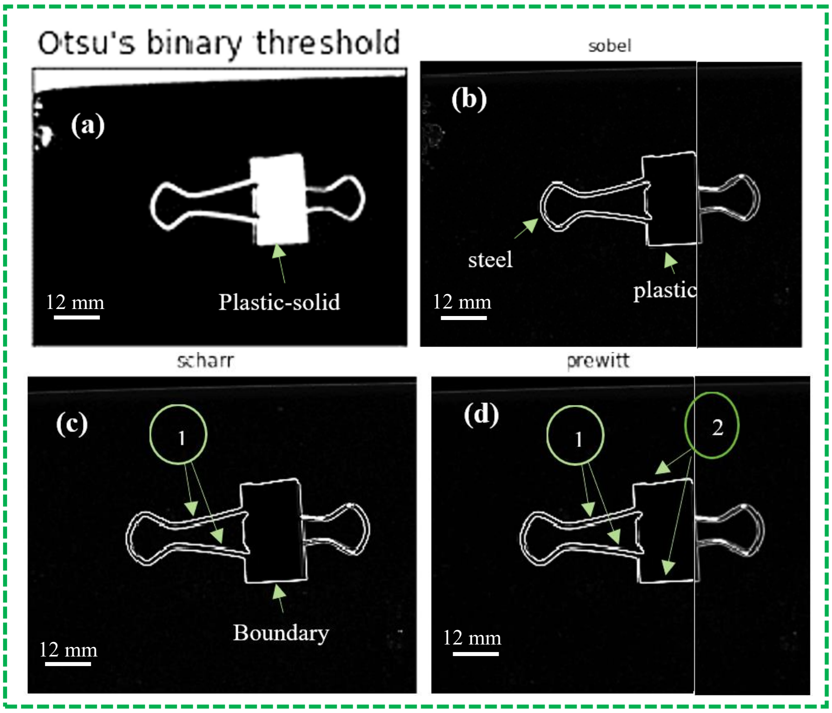

Fig. 6. Unfolding state: a) Otsu's binary threshold image; b) Sobel output; c) Scharr output; d) Prewitt output

3.3.3. Edge Detection of Friction Welding Joint

Through this study, it is expressed that the Python program can be used for analysing the edges/ boundaries of the friction weld joints. The Python program can be used for the interdisciplinary application between computer science and welding engineering. A sample of faying surface modified friction welded joint with wrought aluminium and austenitic stainless steel was prepared with the help of a friction welding machine and the sample was cut into two halves by a wire electrical discharge machining (WEDM) machine. The welding face in a specimen is known as a faying surface. In this study, the tapered-faying surface of AISI304L penetrated AA6063 metal. The WI of the shape faying surface modified is as letter 'V' as in Figure 4c. The image shown in figure 4c was used as the original input image for the edge detection. The output results are given in figures 7a to 7d. The edges of the friction weld joint (Fig. 7c- marked as 1) were detected by all the edge detectors. Figure 7a is the threshold view of the joint, where the faying surfaces and the weld interfaces are seen. But, the AA6063 portion was not seen. The reason may be that the AA6063 portion may be dark due to the poor capturing of the image. There will be some of the drawbacks noted through the shadow of the specimen and the bottom edge of the mould are also identified in all the output images. This is sure because of the non-parallel image capturing. The dissimilar metals are differentiated in figure 7b for ease of understanding. The outer circle is nothing but the Bakelite mould which was used to hold the specimen during the characterisation. Figures 7b, 7c & 7d are the Sobel, Scharr, and Prewitt results respectively. A weld defect (Fig. 7c) was detected in the weld zone of the joint by all the operators. This kind of application is needed for the industries to identify the weld defects. Here, the defect means the incomplete penetration between the metals during the welding process which may lead to poor strength. The Python program can be used in weld quality inspection. The defect's size, shape and measurement are also possible. Further, the weld joint edges and the weld zone boundaries are detected as shown in figure 7d. A light greenish shade colour is available in the AISI304L half of the sample, which is not shown on the AA6063 side. This is due to the light reflectivity and the brightness variation of the input images. Prewitt and Scharr's output images were clear and sharp compared to Sobel's. Similarly, the Python algorithm found the surface damages/defects and the boundary of the FW sample and its WI as shown in figures 7e-7h. Thus, the effect of the milling process on the surface roughness was noted on the machined samples and the performance of FW on WI.

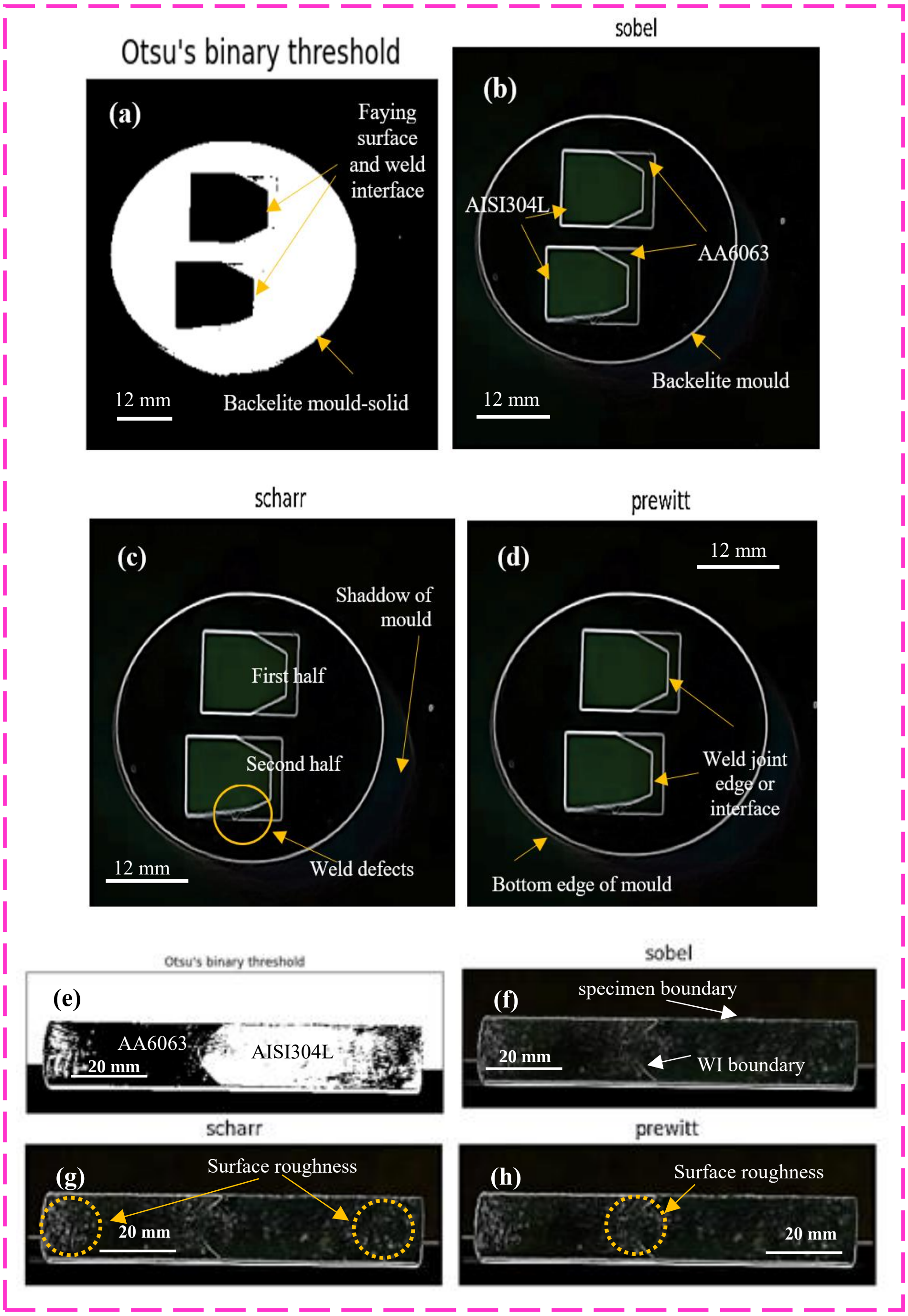

Fig. 7. Friction weld joint sample: a, e) Otsu's binary threshold image; b, f) Sobel output; c, g) Scharr output; d, h) Prewitt output (where images e-h are surface roughness identification on weld sample after milling)

4. CONCLUSIONS

In this particular study, edge detection was focused and an algorithm was successfully crafted and applied to detect edges and defects in two distinct scenarios: a friction welding joint (dissimilar metals) and binder clips (referred to as Case I & II). The flat-faying surfaced AA6063 alloy was successfully joined with a tapered-faying surface of AISI304L alloy by the FW process. The accuracy of the edge detection operators is crucial for tasks such as data extraction and image segmentation in machine vision applications. The algorithm effectively identified the weld interface, weld zone boundaries, and edges of the binder clips. It also demonstrated the capability to detect welding defects near the weld zone. The algorithm produced threshold, Sobel, Scharr, and Prewitt outputs, which were then compared with the original images. This technology finds applicability across a broad spectrum of industries where quality inspection tasks such as defect detection, object boundary analysis, dimensional measurements, surface inspections, pattern recognition, and real-time monitoring of fabricated products are essential. Moreover, this system is versatile and can be employed with objects of varying sizes, shapes, and materials. Among the three operators evaluated in this study, Scharr and Prewitt stand out due to their ability to provide sharp results in both dark and light areas. Looking ahead, the future of edge detection can be characterized by interdisciplinary collaborations, enhanced noise tolerance, adaptive thresholding mechanisms, and integration with deep learning techniques. Future scope can be addressing scenarios with dimly lit images and improving the algorithm's performance in three-dimensional contexts. Python-based edge detection represents a cutting-edge technology employed for the precise identification of object boundaries within captured images.

APPENDIX

Python Coding applied for the Edge Detection

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
from google.colab import drive
drive.mount('/content/drive')
img=
cv2.imread(r'/content/drive/MyDrive/Picture1.jpg')
b,g,r = cv2.split(img)
rgb_img = cv2.merge([r,g,b])
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
ret, thresh =
cv2.threshold(gray,0,255,cv2.THRESH_BINARY_INV
+cv2.THRESH_OTSU) # noise removal
kernel = np.ones((2,2),np.uint8)
opening =
cv2.morphologyEx(thresh,cv2.MORPH_OPEN,kernel
, iterations = 2)
closing =
cv2.morphologyEx(thresh,cv2.MORPH_CLOSE,kerne
l, iterations = 2) # sure background area
sure_bg = cv2.dilate(closing,kernel,iterations=3) #
Finding sure foreground area
dist_transform =
cv2.distanceTransform(sure_bg,cv2.DIST_L2,3) #
Threshold
ret, sure_fg =
cv2.threshold(dist_transform,0.1*dist_transform.max(
),255,0) # Finding unknown region
sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg,sure_fg) # Marker
labelling
ret, markers = cv2.connectedComponents(sure_fg)
# Add one to all labels so that sure background is not
0, but 1 markers = markers+1
# Now, mark the region of unknown with zero
markers[unknown==255] = 0
markers = cv2.watershed(img,markers)
img[markers == -1] = [255,0,0]
plt.subplot(211),plt.imshow(rgb_img)
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(212),plt.imshow(thresh, 'gray')
plt.imsave(r'thresh.png',thresh)
plt.title("Otsu's binary threshold"), plt.xticks([]),
plt.yticks([])
plt.tight_layout()
plt.show()
from skimage.filters import roberts, sobel, scharr,
prewitt
img =
cv2.imread(r'/content/drive/MyDrive/Picture1.jpg')
#edge_roberts=roberts(img)
#plt.imshow(edge_roberts, cmap='gray')
edge_sobel=sobel(img)
edge_scharr=scharr(img)
edge_prewitt=prewitt(img)
fig,axes=plt.subplots(nrows=2,ncols=2,
sharex=True, sharey=True, figsize=(8,8))
ax=axes.ravel()
ax[0].imshow(img, cmap=plt.cm.gray)
ax[0].set_title('original image')
ax[1].imshow(edge_sobel, cmap=plt.cm.gray)
ax[1].set_title('sobel')
ax[2].imshow(edge_scharr, cmap=plt.cm.gray)
ax[2].set_title('scharr')
ax[3].imshow(edge_prewitt, cmap='gray')
ax[3].set_title('prewitt')
for a in ax:
  a.axis('off')
plt.tight_layout()
plt.show()
import numpy as np
import cv2
from matplotlib import pyplot as plt
```

```
img =
cv2.imread('/content/drive/MyDrive/Picture1.jpg')
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.figure(figsize=(8,8))
plt.imshow(img,cmap="gray")
plt.axis('off')
plt.title("Original Image")
plt.show()
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.figure(figsize=(8,8))
plt.imshow(gray,cmap="gray")
plt.axis('off')
plt.title("GrayScale Image")
plt.show()
ret, thresh = cv2.threshold(gray, 0,
255,cv2.THRESH_BINARY_INV
+cv2.THRESH_OTSU)
plt.figure(figsize=(8,8))
plt.imshow(thresh,cmap="gray")
plt.axis('off')
plt.title("Threshold Image")
plt.show()
#for Case II (for image 4.b):
# SEGMENTATION
import numpy as np
import cv2
from matplotlib import pyplot as plt
img =
cv2.imread(r'/content/drive/MyDrive/Picture2.jpg')
b,g,r = cv2.split(img)
rgb_img = cv2.merge([r,g,b])
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
ret, thresh =
cv2.threshold(gray,0,255,cv2.THRESH_BINARY_INV
+cv2.THRESH_OTSU) # noise removal
kernel = np.ones((2,2),np.uint8)
opening =
cv2.morphologyEx(thresh,cv2.MORPH_OPEN,kernel
, iterations = 2)
closing =
cv2.morphologyEx(thresh,cv2.MORPH_CLOSE,kerne
l, iterations = 2) # sure background area
sure_bg = cv2.dilate(closing,kernel,iterations=3)
# Finding sure foreground area
dist_transform =
cv2.distanceTransform(sure_bg,cv2.DIST_L2,3) #
Threshold
ret, sure_fg =
cv2.threshold(dist_transform,0.1*dist_transform.max(
),255,0) # Finding unknown region
sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg,sure_fg) # Marker
labelling
ret, markers = cv2.connectedComponents(sure_fg)
# Add one to all labels so that sure background is not
0, but 1 markers = markers+1
# Now, mark the region of unknown with zero
markers[unknown==255] = 0
markers = cv2.watershed(img,markers)
img[markers == -1] = [255,0,0]
plt.subplot(211),plt.imshow(rgb_img)
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(212),plt.imshow(thresh, 'gray')
plt.imsave(r'thresh.png',thresh)
plt.title("Otsu's binary threshold"), plt.xticks([]),
plt.yticks([])
plt.tight_layout()
plt.show()
from skimage.filters import roberts, sobel, scharr,
prewitt
img =
cv2.imread(r'/content/drive/MyDrive/Picture2.jpg')
#edge_roberts=roberts(img)
#plt.imshow(edge_roberts, cmap='gray')
edge_sobel=sobel(img)
edge_scharr=scharr(img)
edge_prewitt=prewitt(img)
fig,axes=plt.subplots(nrows=2,ncols=2,
sharex=True, sharey=True, figsize=(8,8))
ax=axes.ravel()
ax[0].imshow(img, cmap=plt.cm.gray)
ax[0].set_title('original image')
ax[1].imshow(edge_sobel, cmap=plt.cm.gray)
ax[1].set_title('sobel')
ax[2].imshow(edge_scharr, cmap=plt.cm.gray)
ax[2].set_title('scharr')
ax[3].imshow(edge_prewitt, cmap='gray')
ax[3].set_title('prewitt')
for a in ax:
  a.axis('off')
plt.tight_layout()
plt.show()
import numpy as np
import cv2
from matplotlib import pyplot as plt
img =
cv2.imread('/content/drive/MyDrive/Picture2.jpg')
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.figure(figsize=(8,8))
plt.imshow(img,cmap="gray")
plt.axis('off')
plt.title("Original Image")
plt.show()
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.figure(figsize=(8,8))
plt.imshow(gray,cmap="gray")
plt.axis('off')
plt.title("GrayScale Image")
plt.show()
ret, thresh = cv2.threshold(gray, 0,
255,cv2.THRESH_BINARY_INV
+cv2.THRESH_OTSU)
plt.figure(figsize=(8,8))
plt.imshow(thresh,cmap="gray")
plt.axis('off')
plt.title("Threshold Image")
plt.show()
```

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Center for Machine Vision, Industry Institute Interaction Cell (IIIC), Rajalakshmi Engineering College (Autonomous), Chennai for their support in learning Python.

REFERENCES

[1] https://www.science.gov/topicpages/m/machine+vision+research

[2] **Mishra A.**, *Understanding Machine Learning for Friction Stir Welding Technology*, 2020, pp. 1-17, http://dx.doi.org/10.2139/ssrn.3560731.

[3] https://qualitastech.com/blog/quality-control-insights/9-common-applications-of- machine-vision-in-manufacturing/.

[4] **Ambrosio D., Wagner V., Dessein G., Vivas J., Cahuc O.**, *Machine Learning Tools for Flow-Related Defects Detection in Friction Stir Welding*, Journal of Manufacturing Science and Engineering, Vol. 145(10), 101005, 2023, https://doi.org/10.1115/1.4062457.

[5] **Arun Bhatia S.**, *Review on Edge Detection Techniques for Pharmaceutical Drugs*, International Journal for Scientific Research & Development, 2016, Vol. 4(1), pp. 4-7.

[6] **Mishra A.**, *Artificial Intelligence Algorithms for the Analysis of Mechanical Property of Friction Stir Welded Joints by using Python Programming*, Welding Technology Review, Vol. 92(6), 2020, https://doi.org/10.26628/wtr.v92i6.1120.

[7] **Mishra A., Bhatia Dutta S.**, *Detection of surface defects in friction stir welded joints by using a novel machine learning approach*, Applied Engineering Letters, Vol.5(1), 2020, pp: 16-21, https://doi.org/10.18485/aeletters.2020.5.1.3

[8] https://en.wikipedia.org/wiki/Binder_clip.

[9] US1865453A, *Binder Clip*, US patent, 1932.

[10] **Kejin Z., Xusheng Q., Jieshi C., Junmei C., Hao L**, *Non-monotonic evolution of microstructure and fatigue properties of round bar–plate rotary friction welding joints in 304 austenitic stainless steel*, Materials & Design, Vol. 224, 111400, 2022 https://doi.org/10.1016/j.matdes.2022.111400.

[11] **Fua P., Leclerc Y. G.**, *Model driven edge detection. Machine Vis. Apps.* 3, 1990, pp: 45–56, https://doi.org/10.1007/BF01211451

[12] Bhardwaj S., Mittal A., *A Survey on Various Edge Detector Techniques, Procedia Technology*, Vol. 4, pp. 220-226, 2012, https://doi.org/10.1016/j.protcy.2012.05.033.

[13] **Vinista P., Milton M., Joe A.,** *Novel Modified Sobel Algorithm for Better Edge Detection of Various Images*, International Journal of Emerging Technologies in Engineering Research, Vol. 7(3), 2019, pp. 25-31.

[14] **Muthukrishnan. R, Radha M.**, *Edge Detection Techniques for Image Segmentation*, International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3(6), 2011.

[15] **Mishra A.,** *Image Processing of Friction Stir Welded 6060-T5 Aluminum Alloy Joint*, Journal of Aircraft and Spacecraft Technology, Vol. 4, 2020, DOI: 10.3844/jastsp.2020.48.53.

[16] **Senthil M. S., Noorul H. A., Sathiya P.**, *Eco-friendly frictional joining of AA6063 and AISI304L dissimilar metals and characterisation of bimetal joints*, Journal of New Materials for Electrochemical Systems, Vol. 23(2), 2020, pp. 101-111. https://doi.org/10.14447/jnmes.v23i2.a07.

[17] **Senthil Murugan S., Noorul Haq A., Sathiya P.,** *Effect of welding parameters on the microstructure and mechanical properties of the friction-welded dissimilar joints of AA6063 alloy and faying surface-tapered AISI304L alloy,* Welding in the World, Vol. 64, 2020, pp. 483–499, https://doi.org/10.1007/s40194-020-00846-x.

[18] **Senthil Murugan S., Sathiya P., Noorul Haq A.**, *Experimental study on the effect of silver, nickel and chromium interlayers and upset pressure in joining SS304L-AA6063 alloys through direct drive friction welding process,* Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. 42, 611, 2020, https://doi.org/10.1007/s40430-020-02687-7.

[19] **Senthil Murugan S., Sathiya P., Noorul Haq A.**, *Rotary Friction Welding and Dissimilar Metal Joining of Aluminium and Stainless Steel Alloys,* Annals of “Dunarea de Jos” University of Galati. Fascicle XII, Welding Equipment and Technology, Vol. 32, 2021, pp. 85-92.