Received: August 24, 2023. Revised: December 1, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhancing GLCM-Based Feature Extraction for THT Solder Joint Inspection through Log Polar Transformation

Eko Purwanto Aribowo[1]* **Albertus Joko Santoso[2]** **Pranowo[2]**

[1]*Department of Mechatronics, Politeknik ATMI Surakarta, Indonesia*
[2]*Department of Informatic, Faculty of Industrial Technology, University of Atma Jaya Yogyakarta, Indonesia*
* Corresponding author's Email: eko.purwanto@atmi.ac.id

Abstract: Quality inspection of solder connections in electronic circuit manufacturing is commonly performed using automatic optical inspection (AOI) technology. The utilization of deep learning in AOI has demonstrated high accuracy and fast computation, yet it requires expensive graphics processing unit (GPU) equipped computers. To enable broader and more cost-effective utilization, there is a need for an AOI model that can be embedded in simple central processing unit (CPU) based computers. To pursue this objective, several research efforts have been undertaken to develop AOI models based on classical machine learning techniques. However, the accuracy and speed achieved by these models have not yet matched deep learning-based AOI models. This study aims to enhance the computational efficiency of classic machine learning by processing only pixels containing textural information. The effectiveness enhancement is achieved through the application of log-polar transformation in the extraction of texture features using the gray level co-occurrence matrix (GLCM) to detect defects in trough hole technology (THT) solder joint connections. By transforming cartesian coordinates into polar coordinates, the textural areas to be detected assume a square shape, facilitating efficient texture feature extraction. To ensure a significant improvement in performance, a comparative performance evaluation is conducted on classic machine learning-based AOI models with and without the log-polar transformation. The texture features extracted from both models are classified using the support vector machine (SVM) method. Model testing and evaluation reveals that the proposed enhancement effort is capable of increasing accuracy levels to 95% and reach computation time by 120 milliseconds.

Keywords: Solder joint inspection, THT, GLCM, Log-polar transformation, Performance evaluation.

1. Introduction

Solder joint inspection is a critical step in the electronic manufacturing process. Poor solder joints can result in product damage and even pose risks to human safety. Various defects in solder joints are caused by the difficulty in maintaining accuracy in mechanical and chemical parameters during the soldering process [1, 2]. Due to the various risks of defects in electronic circuit manufacturing, solder joint inspection becomes an essential aspect to ensure that the produced products have high-quality solder joints and meet the required standards [3].

Automated quality testing of solder joints is necessary due to the high production volume and short production time, making manual testing impractical. Automatic optical inspection (AOI) can accurately and efficiently identify defects in products without being influenced by human subjectivity, thus reducing production time and costs [4, 5]. Research conducted by Ebayyeh [6] demonstrates that AOI can help improve the quality of solder joints and reduce defects in electronic products. Another study [7] shows that AOI can identify and classify solder joint defects with high accuracy. As a result, AOI has become an increasingly popular technology in the electronic industry due to its vital role in ensuring the quality of electronic products. In AOI-based quality inspection of solder joints, there are two data processing methods that can be utilized: classical machine learning and deep learning. Both methods

Received: August 24, 2023. Revised: December 1, 2023.

Table 1. Algorithm performance comparison

Method		Accuracy	Speed
Deep learning	VGG16 [20]	90%	
	YOLOV3[21]	95%	0.33s
	CNN [22]	96%	
	CNN [23]	90%	
	CNN [24]	95%	
	YOLO V5	97.5%	0.1s
Classical Machine Learning	k-NN (k=4)	87%	0.39s
	k-NN (k=1) with PCA	67%	0.28s
	SVM	81%	0.13s
	Decicion Tree	73%	0.75s

have their own strengths and weaknesses [6-8]. AOI models using deep learning methods can provide more accurate results in detecting solder joint defects but require a larger amount of data and longer processing time compared to classical machine learning techniques.

Although deep learning techniques have shown impressive performance, their reliance on costly graphics processing unit (GPU) hardware limits their availability to large-scale electronics manufacturers. On the other hand, more affordable AOI technology can cater to a broader user base. The use of microcomputers also allows for the creation of compact-sized devices, making the technology accessible to small-scale electronics manufacturers, electronics technicians, maintenance teams, and even individual users.

Considering these opportunities, several studies [9-19] have aimed to develop algorithms capable of running on lower-capacity CPU-based computers. The approach involves utilizing classical machine learning techniques to ensure that the algorithm can run on simple CPU devices, like microcomputers or single-board computers. Nevertheless, it's important to continually compare the performance of the AOI models developed with the performance of commonly used deep learning AOI models.

Table 1 illustrate the computational performance comparison in automated optical inspection (AOI) using deep learning and classical machine learning models.

In Table 1, the computational performance achieved using deep learning models shows accuracy ranging from 90% to 97.5% and processing speeds of 0.1 to 0.33 seconds. Research conducted by [19] aimed to construct classical machine learning models that approach the performance of deep learning. However, while the computational performance obtained has not yet surpassed that of deep learning methods, it is not an indication of failure in the research endeavors. In fact, there exists untapped potential for optimizing the development of machine learning-based models. One issue that has yet to be addressed is the computationally expensive and inefficient image processing when dealing with empty pixel data on THT solder joint processing.

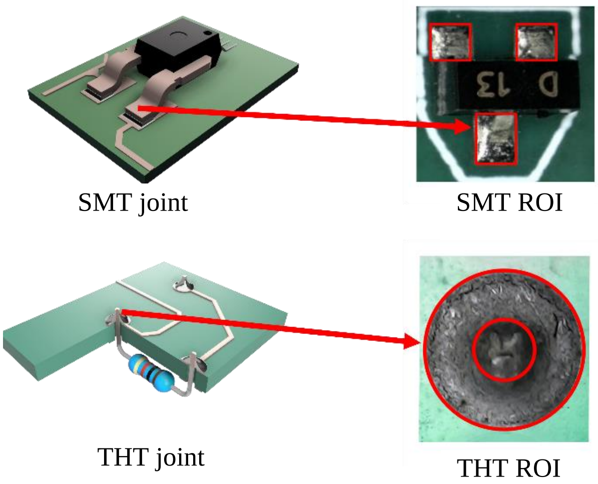


Figure. 1 Printed circuit board mounting schemes

Previous research has yielded good accuracy in assessing the quality of solder joints with surface mounted technology (SMT) types. However, the accuracy decreases when applied to trough hole technology (THT) solder joints due to their circular shape [25]. This conundrum poses a significant challenge in the realm of computational analysis. Performing meticulous computations typically demands an extended processing time. In contrast, prioritizing swiftness can potentially compromise the accuracy of defect detection. Hence, there arises a pressing need for methodologies that can achieve precise defect detection within stringent computational time constraints. Fig. 1 illustrates the solder joint structures of both types.

In surface mounted technology (SMT), the solder joint structure, as depicted in the upper left of Fig. 1, comprises square-shaped pads that match the shape of the component pins. These solder joints are specifically designed to align with the component's pin configuration. The quality testing of SMT solder joints involves creating a region of interest (ROI) within the solder joint area. On the other hand, the solder joints in through hole technology (THT) have a conical shape, as shown in the lower left of Fig. 1, resulting in a circular ROI. The central part of the solder joint image corresponds to the component pin, which is not included in the analysis. Thus, the processed ROI will take the form of a ring.

The application of the co-occurrence matrix is typically designed for square-shaped frames, while segmented images often have circular shapes. This

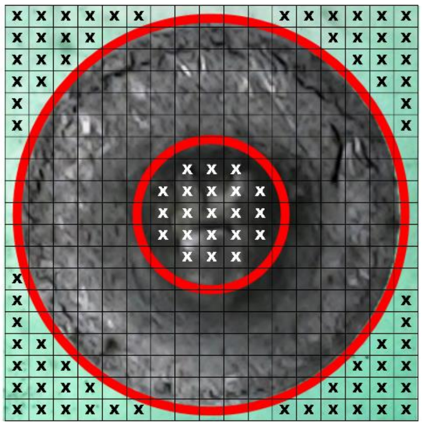


Figure. 2 THT solder joint image with irrelevant pixels

creates a challenge where a number of pixels that are unnecessary for analysis are included in the co-occurrence calculations, as depicted in Fig. 2.

In Fig. 2, the image illustrates a THT solder joint with square-shaped overlay layers representing pixel interpretation. The "x" marks outside the segmented image indicate pixels that are not relevant for texture analysis and can be excluded from further processing. The presence of irrelevant pixels included in the GLCM operation diminishes the accuracy of the model. However, this issue has not been resolved in previous studies [9-19], prompting the need for a solution in this research.

The author recognizes the substantial potential for optimizing the accuracy of solder joint detection through classical machine learning by implementing the log-polar transform method. Therefore, the principal objective of this research is to enhance the previously proposed methods to achieve defect detection in THT solder joints with an accuracy ranging between 90% and 95.7% and processing speeds within the range of 0.33 to 0.1 seconds. This target accuracy and speed range is typically associated with deep learning-based models.

The following sections of this study are structured as follows: Section 2 provides a relevant overview of the current research. Section 3 encompasses the methodology employed in the study. Finally, Sections 4 and 5 present in-depth performance measurement metrics and a comparative analysis of the results.

2. Related works

The primary challenge addressed in this research pertains to enhancing the accuracy of through hole technology (THT) solder joint detection through classical machine learning approaches. The objective is to develop algorithms capable of running on microcomputers while maintaining a level of accuracy comparable to deep learning-based AOI systems widely employed in the industry. To achieve this objective, it is imperative to conduct an in-depth exploration of related works, with a specific focus on defect detection using classical machine learning and texture feature extraction.

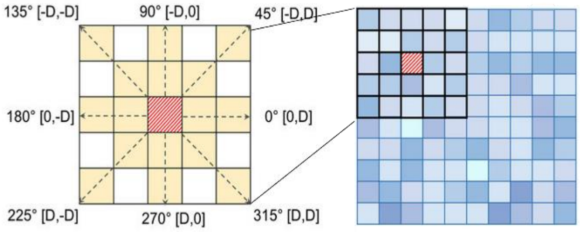


Figure. 3 Illustration of how a kernel window works to perform GLCM

Among the commonly employed feature extraction techniques in machine learning based AOI, the gray level co-occurrence matrix (GLCM) stands out as one of the prominent methods. Segmentation is required in texture feature extraction utilizing GLCM [20]. Furthermore, the extracted GLCM features are then classified using machine learning techniques, such as support vector machine-based classification [26]. In the study [17, 27], a pre-trained machine learning model incorporating GLCM-based texture feature extraction achieved remarkable results. The AOI model with GLCM-based feature extraction exhibits excellent accuracy in detecting solder joints of the surface mounted technology (SMT) type. However, its accuracy diminishes when applied to through-hole technology (THT) solder joints.

GLCM operates using a kernel window, as mentioned by Haralick [28]. The kernel window in GLCM analysis compares the pixel values within a specified neighborhood to calculate the occurrence and distribution of different pixel value pairs [29]. With this kernel, GLCM captures the patterns of correlation between pixels in the inspected image and is commonly employed to identify texture features in images [30]. Study [31] illustrated the operation of the kernel window on the image as depicted in Fig. 3.

However, the circular shape of THT solder joints does not align with the shape and direction of kernel window movement, leading to the inclusion of unintended pixels in the computation, as illustrated in Fig. 2. In several studies, the focus has primarily been on surface-mounted technology (SMT) solder joints using square-shaped solder joint images. This limitation has resulted in less accurate defect detection models when applied to through-hole technology (THT) solder joints [9-19].

In various texture analysis cases, the combination of texture analysis techniques with other image processing methods is necessary to achieve the desired effectiveness. Texture features can be defined using descriptors and can be obtained through operations involving single or combined models for texture analysis [32]. For example, the combination of Zernike transformation and GLCM was employed [33] to overcome the limitations of co-occurrence matrices. The research findings supported the effectiveness of the proposed approach in detecting contrast enhancement when forensic fingerprints are tampered with anti-forensic attacks, achieving a true positive rate (TPR) of 92.0%. Another study [34] utilized an approach that combined the rectangular-to-polar coordinate transformation, previously introduced by [35], with GLCM. The proposed model in their study demonstrated improved accuracy and computational speed compared to the classical GLCM method.

Referring to [32] and the successful research [34], the combination of polar transformation and GLCM feature extraction shows promising potential when applied to circular-shaped objects. This is because the integration of polar transformation with GLCM allows for the modeling of texture that not only considers the spatial relationships between neighboring pixels in the image but also considers the radial representation of the texture. This provides additional information about the direction and pattern of the texture in the image [36, 37].

One of the primary innovations introduced in this research is the utilization of the log-polar transformation in the extraction of texture features from THT solder joints. The log-polar transformation offers several significant advantages compared to conventional methods. With this approach, unnecessary processing of irrelevant pixels in GLCM calculations can be avoided, resulting in improved computational efficiency. Furthermore, the log-polar transformation enables us to achieve higher detection accuracy, particularly when dealing with complex-shaped THT solder joints, as illustrated in Fig. 2.

3. Methodology

This section will specifically discuss the proposed method in this research, as illustrated in Fig. 4. The research procedure is divided into five stages: the pre-processing stage, segmentation, log-polar transformation, feature extraction, and classification. Furthermore, a performance evaluation step will be conducted using a confusion matrix to demonstrate that the use of log-polar transformation has a significant impact on improving the accuracy of solder joint defect detection based on GLCM texture feature extraction.

Fig. 4 illustrates the five stages of the research methodology, starting with Stage 1: Preprocessing. This stage consists of image acquisition, conversion of the image format from RGB to grayscale, and generation of the binary image. Stage 2 is segmentation, which includes operations such as active contouring, convex hull, central point detection, and region of interest generation.

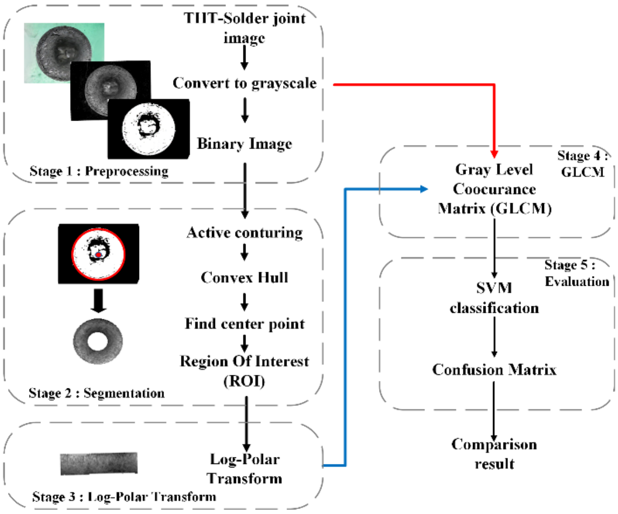


Figure. 4 Research methodology

Stage 3 focuses on the log-polar transformation, which is the central aspect of this research. The output of the log-polar transformation is a rectangular image, indicated by the blue arrow at Fig. 4, and its features are extracted using the GLCM (gray level co-occurrence matrix) method. The red arrow (Fig. 4) represents the grayscale image without the log-polar operation, and its features are also extracted using GLCM.

The feature data from both images are then classified using SVM (support vector machine) to obtain the defect and normal classes. To evaluate the performance of the two detection models, a confusion matrix is used. This process allows for a performance comparison between feature extraction using log-polar transformation and feature extraction without log-polar transformation in detecting THT (through-hole technology) solder joint defects.

By following these stages, a performance comparison can be made between feature extraction techniques using log-polar transformation and those without log-polar transformation in the detection of THT solder joint defects.

3.1 Preprocessing

During the image acquisition stage, images are captured in the RGB format, comprising three color channels: red (R), green (G), and blue (B). In further image processing, calculations are carried out on a single channel of pixel intensity. Hence, it is essential to convert the image format from RGB to grayscale to focus solely on the grayscale channel for subsequent analysis and operations. To obtain a grayscale image with a single-color channel, a conversion process is required. This conversion involves calculating the combination of information from the three-color channels into a single grayscale channel. The aim is to merge the color information and generate a grayscale image that represents the overall brightness or intensity of the original RGB image.

The approach used in this research is to calculate the weighted average of pixel values for each color channel. This calculation considers the relative contributions of each color channel in forming the grayscale image, as illustrated in Eq. 1.

$$Gry = W1\ \times R + W2\ \times G + W3 \times B \tag{1}$$

Where *Gry* = Single channel pixel intensity at coorditate (y, x) picture. *R* = Red channel, *G* = Green channel, and *B* = Blue channel intensity at cooriditate (y, x). The constants W1, W2, and W3 are used in the conversion process, where W1 = 0.2989, W2 = 0.5870, and W3 = 0.1140. These constants represent the weights that indicate the relative contributions of the color channels in the RGB to grayscale image conversion. Specifically, these weights adhere to the Rec. 709 standard, which is widely utilized in image processing and color reproduction to achieve accurate color representation. This operation transforms each pixel in the RGB image into a single pixel value in the grayscale image, reflecting the level of brightness or intensity of each pixel.

3.2 Segmentation

The resulting grayscale image is then processed using a thresholding operation to obtain a binary image. In the thresholding operation, a group of pixels is categorized into either the foreground or the background based on the desired segmentation. This is achieved by comparing the intensity values of each pixel in the image with a predetermined threshold value. Pixels that meet certain conditions are assigned values that indicate whether they belong to the foreground or the background in the binary image. Mathematically, the thresholding operation can be expressed using Eq. (2).

$$b(y,x) = f(x) = \begin{cases} 1, for\ (y,x) \geq T \\ 0, for\ (y,x) < T \end{cases} \tag{2}$$

Where *b* represents pixel on cooriditate (y, x) and T represents treshold value. The threshold value is determined through empirical evaluation of multiple datasets to obtain an optimal value. Once the optimal threshold value, denoted as T, is obtained, subsequent evaluations yield relatively consistent results as the evaluation environment and camera equipment remain unchanged. The thresholding operation transforms each pixel in the image into a value of 1 or 0, depending on the pixel's intensity value. Specifically, in Eq. (2), the pixel value at coordinate (y, x) is set to 1 if its intensity value (f(x)) is greater than or equal to the threshold value T, and it is set to 0 if the intensity value is smaller than the threshold value T.

The resulting binary image is essential for the active contour operation using the snake method, where the contour of the desired object is iteratively adjusted to fit its boundaries. The snake method employs energy minimization principles to determine the optimal contour that aligns with the object's edges or features in the image. By leveraging the binary image that distinguishes the foreground (object) from the background, the snake algorithm efficiently converges towards the object's boundaries, enabling accurate segmentation.

However, the snake operation alone may not provide accurate centroid detection for the object. Therefore, the convex hull technique is employed to find the smallest polygon from the image. By incorporating the convex hull technique, centroid determination remains unaffected by high levels of noise that might not be handled effectively by the snake operation. This combined approach of the snake operation and convex hull technique yields improved accuracy in determining the object's centroid in the image. To perform the convex hull calculation, it is necessary to set X10 = A, where A represents the image to be computed in Eq. (3). This equation will converge on X_k^{i} = X_(k-1)^{i}, indicating the iterative process for computing the convex hull.

$$X_k^i = thm(X_{k-1}, B^i) \cup A \tag{3}$$

Eq. 3 represent a convex set that encompasses all points connecting two points within the set. It is also referred to as the smallest polygonal shape that can enclose an object. The computation of the convex hull involves the application of the "Hit_or_Miss" transformation with a 90° rotated structural element. The variables i and k represent the iterations within the convex hull operation.

The segmentation stage is further followed by computing the centroid of the convex hull. This centroid plays a crucial role in determining the center of the circle during the segmentation process. The

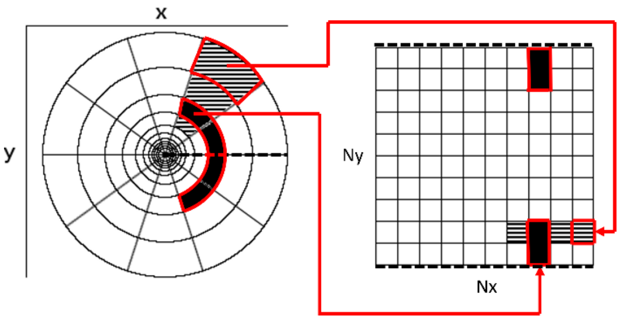


Figure. 5 Log-polar transformation

segmented image is then formed as a ring with inner and outer diameters. The inner diameter of the ring is calculated as d = 1/8 D, while the outer diameter is represented as D = DX, where DX represents the diameter of the convex hull.

3.3 Log-polar transform

In this operation, the radial coordinate is transformed according to the rule: r = e^{ρ}. In this rule, r represents the radial coordinate in the Log-Polar representation, which represents the radial distance in the log-polar coordinate system shown in Eq. (4).

$$\rho = ln\sqrt{y^2 + x^2} \quad (4)$$

To calculate the value of ρ, the natural logarithm (ln) function is used, which takes the argument as the result of $\sqrt{y^2 + x^2}$. The natural logarithm (ln) function is mathematically defined as the inverse of the exponential function with the base e. In Eq. (4), the ln function is used to compute the exponent value that utilizes the base e (approximately 2.71828). In other words, the ln function seeks the exponent value that satisfies the equation $e^{\rho} = \sqrt{y^2 + x^2}$.

The Log-Polar transformation involves applying Eq. (4), which utilizes the natural logarithm (ln) function to convert the radial distance value from the Cartesian coordinate system to a logarithmic representation in the log-polar coordinate system. This transformation helps in mapping the radial distances to a logarithmic scale, facilitating better representation and analysis of the data in the Log-Polar domain [42]. This representation is shown in Fig. 5.

Thus, the exponent value for the corresponding radial distance value can be obtained using the natural logarithm (ln) function, which utilizes the base e. The resulting image from the Log-Polar transformation takes the form of the function f (x, y), with dimensions of width Nx = D-d and Ny = K, where K represents the circumference of the outer diameter of the circle.

3.4 Gray level co-ocurance matrix (GLCM)

GLCM (gray-level co-occurrence matrix) is a method for analyzing texture features with two orders. First-order texture measurements involve statistical calculations based on the original pixel values of an image, such as variance, and disregard the spatial relationships between neighboring pixels. In second-order calculations, the relationships between pairs of original pixels in the image are considered, examining the co-occurrence patterns of gray-level values. The image obtained from the log-polar transformation consists of spatial direction vectors (r) ⃗ and intensity levels L. [GLCM]_r ⃗ (i,j) represents the count of pixels with a value of j ∈1..., L that appear at an offset (r) ⃗ from a pixel with a value of i ∈1..., L. The GLCM utilized in the analysis can be expressed using Eq. (5).

$$GLCM_{\vec{r}}(i,j) = \#\left\{(x_1, y_1), (x_2, y_2) \in \left(N_x, N_y\right) \times \left(N_x, N_y\right) | f(x_1, y_1) = j^{\vec{r}} = \overrightarrow{(x_2 - x_1, y_2 - y_1)}\right\} \quad (5)$$

In this study, GLCM (gray-level co-occurrence matrix) is employed to analyze the texture patterns in images obtained from the log-polar transformation. GLCM provides information about the frequency of occurrence of pixel pairs with specific intensity values at predetermined distances and directions. Feature extraction from GLCM yields four feature data: Disimilarity, energy, entropy, and correlation. These features capture different aspects of the texture patterns and contribute to the characterization and analysis of the log-polar transformed images. Each feature is calculated mathematically using the respective equations. Here are the mathematical equations for each feature:

$$Disimilarity = \sum_{i=1}^{L} \sum_{j=1}^{L} |i - j| (GLCM(i,j) \quad (6)$$

$$Energy = \sum_{n=1}^{L} n^2 \left\{\sum_{|i-j|=n} GLCM(i,j)\right\} \quad (7)$$

$$Entropy = -\sum_{i=1}^{L} \sum_{j=1}^{L} (GLCM(i,j) \log (GLCM(i,j) \quad (8)$$

$$Correlation = \frac{\sum_{i=1}^{L} \sum_{j=1}^{L} (ij)(GLCM(i,j) - \mu_i{}' \mu_j{}'}{\sigma_i{}' \sigma_j{}'} \quad (9)$$

With

$$\mu_i{}' = \sum_{i=1}^{L} \sum_{j=1}^{L} i * GLCM(i,j)$$

$$\mu_j{}' = \sum_{i=1}^{L} \sum_{j=1}^{L} j * GLCM(i,j)$$

$$\sigma_j{}^2 = \sum_{i=1}^{L}\sum_{j=1}^{L} GLCM(i,j)\big(j-\mu_j{}'\big)^2$$

$$\sigma_i{}^2 = \sum_{i=1}^{L}\sum_{j=1}^{L} GLCM(i,j)(i-\mu_i{}')^2$$

(i, j) represents the GLCM matrix elements at position (i, j). The summation operation is then applied across all elements of the GLCM matrix. μ_i and μ_j are the mean gray-level intensities of pixels i and j in the GLCM matrix, respectively. (i, j) denotes the probability of occurrence of pixel pairs with gray-level intensities i and j. σ_i and σ_j represents the standard deviations of gray-level intensities for pixels i and j in the GLCM matrix, respectively.

3.5 Classifications

The classification process on the feature data obtained through GLCM texture feature extraction is performed using a support vector machine (SVM) classification model. The classification process begins with training the SVM classification model. Weighted training data is required to perform the training, which yields the values, w, and the constant, b, to obtain the optimal hyperplane. The weighted training data is converted into vector form with values y ∈ (-1, 1). The largest margin obtained through calculations using Eq. (10) is used with constraints expressed by inequality (11). The mathematical model of the problem can be solved more easily by converting it into a Lagrange function and simplifying it into Eq. (12), with α_i ≥ 0 as the Lagrange multiplier. A linear kernel is used to separate the data as shown in Eq. (13).

$$Minimize\ = \frac{1}{2}\ ||W||^2 \tag{10}$$

$$y_i(w \times x_i\ + b) - 1 \geq 0, \forall i \tag{11}$$

$$L(w,b,a) = \frac{1}{2}\ ||W||^2 - \sum_{i=1}^{n} a_i\, y_i(w^T \times x_i\ + b) + \sum_{i=1}^{n} a_i \tag{12}$$

$$K\big(x_i, x_j\big) = \big(x_i^T, x_j\big) \tag{13}$$

After the training is completed, the testing phase is conducted to evaluate the performance of the SVM classification model. Test data, which has also undergone feature extraction from the GLCM matrix, is used to make predictions on class labels based on the previously trained model. The SVM model will attempt to separate the test data based on the hyperplane constructed during the training phase and classify the data into the appropriate classes.

The classification results from both models yield predictions for each image, categorizing them as either "defect positive" or "defect negative". The performance of each model is then evaluated using a confusion matrix. Based on the classification results, performance metrics such as accuracy, precision, sensitivity, and F1-score are calculated. Accuracy measures how well the models predict correctly, precision assesses the models' ability to accurately classify "defect positive" instances, sensitivity evaluates the models' capability to detect a significant number of true "defect positive" cases, and F1-score provides a comprehensive measure by harmonizing precision and sensitivity. The comparison of the two models' performances can be easily observed through the evaluation metrics provided in the confusion matrix.

Using the classification outcomes from both models, evaluation metrics including accuracy, precision, sensitivity, and F1-score are computed. Accuracy reflects the models' overall correctness in predictions, while precision measures their capability to accurately classify instances as "defect positive". Sensitivity assesses the models' ability to correctly detect a substantial number of true "defect positive" cases, and the F1-score provides a comprehensive overview by considering the balance between precision and sensitivity. The performance comparison between the two models can be readily observed through the evaluation metrics derived from the confusion matrix, providing insights into their accuracy, precision, sensitivity, and F1-score in classifying solder joint images.

4. Result and discussion

In this section, we will discuss the test results of each step leading to the comparison of accurate model performance.

4.1 Pre-processing

The process of converting an RGB image to a grayscale image has been successfully performed using the weighted average calculation method with the Rec 709 standard. This approach combines information from the red, green, and blue color channels with weights specified in the Rec 709 standard, as illustrated in Figure 6. The testing conducted has resulted in a grayscale image with a single-color channel. The successful grayscale conversion enables a simpler representation of the image while still preserving crucial information regarding brightness levels and pixel intensities.

To achieve a grayscale image, a series of operations are executed on the image, as demonstrated in Fig. 6. Initially, the image is

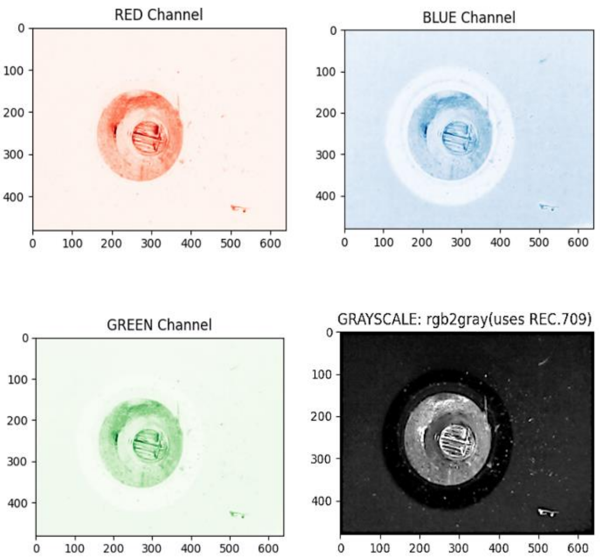


Figure. 6 Gray scale transform

extracted from the red color channel (R), followed by retrieval from the blue color channel (B) and the green color channel (G). Subsequently, each pixel within the color channels undergoes multiplication by the weights prescribed in the Rec 709 standard. The red color channel (R) is multiplied by a weight of 0.2989, the green color channel (G) is subjected to a weight of 0.5870, and the blue color channel (B) is scaled by a weight of 0.1140.

Following this, the outcomes of the multiplicative operations performed on each color channel are aggregated. This summative process yields the grayscale image illustrated in the lower right corner of Fig. 6, wherein each pixel represents the luminance or intensity value, derived from the red (R), green (G), and blue (B) color channels, with weighting in accordance with the Rec 709 standard. This process considers the human eye's sensitivity to red, green, and blue colors, resulting in a grayscale image that accurately represents the light intensity in the image based on the established standard.

4.2 Segmentation

For a more focused image analysis and processing, it is imperative to employ segmentation to separate the relevant regions from the background or extraneous elements. The grayscale image is then subjected to a thresholding operation using Eq. (2). The sequential stages of the thresholding operation in image processing are depicted in Fig. 7.

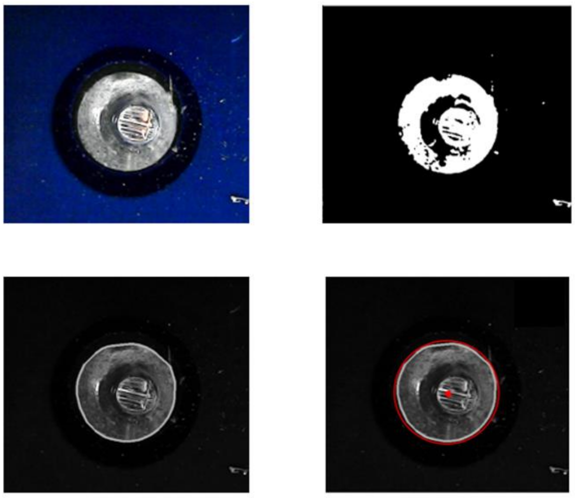


Figure. 7 Segmentation result

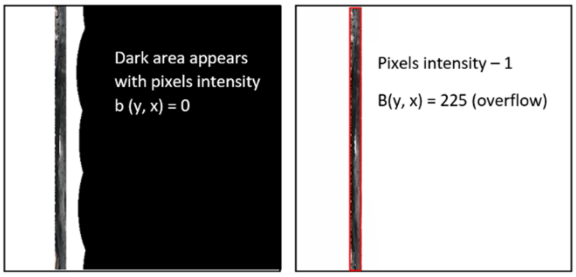


Figure. 8 Log-polar transformation result

The binary image is further processed to enable contour feature operations on the image. To extract contours from the image, a convex hull operation is applied using Eq. (3). The result of the convex hull operation is the morphological contour of the solder joint, which is useful for identifying the centroid and perimeter of the object. After finding the centroid point, the centroid point is used as a reference for cropping the segmented ring-shaped image. The segmented image is then processed with a log-polar transform.

4.3 Log-polar transform

Through the segmentation process, the region of interest has been identified for feature extraction. However, a challenge arises when attempting to apply texture feature extraction to the segmented image, as the feature extraction is typically performed on images with a square-shaped frame, while the segmented region exhibits a ring-like shape. This discrepancy hinders the optimal extraction of texture features. To address this issue, a log-polar transformation is proposed to convert the ring-shaped image into a square-shaped image in Fig. 8.

The log-polar transformation is executed using Eq. (5) to acquire the transformed image, as illustrated in the left portion of Fig. 8. The center of the circle is determined as the transformation center, which is obtained through the centroid search operation in the segmentation stage. Next, the range of the transformation angle is determined, encompassing angles from 0 to 360 degrees. The log-polar transformation process involves the utilization of Eq. (5), which incorporates logarithmic and angle-to-Cartesian coordinate conversions.

Finally, the resulting log-polar transformed image in a square shape is displayed for further visualization and analysis. The log-polar transformation enables a more effective and appropriate representation in a square format, facilitating more efficient processing and manipulation of circular patterns. However, during this stage, the log-polar transform operation includes pixels that are outside the frame. These pixels have a value of 0, which introduces artifacts in the transformed image.

The artifacts manifest as black wavy edges in certain parts of the image. The pixels outside the image frame have a pixel value of 0, so when these pixels are processed, they result in black areas. To address this issue, a simple step is applied by converting the image to an 8-bit integer format and subtracting 1 from all pixels. As a result, an overflow occurs in the pixels that were originally 0, causing them to have a value of 255 or appear as white pixel.

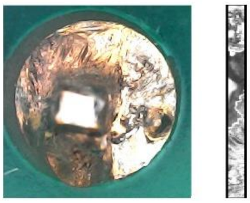
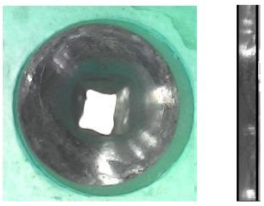
4.4 The comparison between classical GLCM and log-polar GLCM

In this section, we will discuss the comparison of defect detection performance in solder joints using GLCM feature extraction, both with and without the application of log-polar transform. The discussion will begin by examining the performance of GLCM texture feature extraction with log-polar transform. Table 2 demonstrates the results of texture feature extraction for two classes of solder joints. On the left side of the table are solder joints with the disturbed soldering defect class, while on the right side of the table are normal solder joints.

These differences can be observed in terms of dissimilarity, entropy, and correlation, while the feature energy still indicates some differences, although not significant. The analysis of dissimilarity feature reveals that defective solder joints have a higher level of dissimilarity compared to normal solder joints. Defective solder surfaces tend to exhibit larger variations between pixels, creating a rough and irregular texture. In contrast, normal solder joints have smoother surfaces, which can be seen from their lower dissimilarity values.

Regarding the entropy feature, defective solder joints exhibit higher variations in pixel intensities compared to normal solder joints. This is reflected in the difference in entropy values between the two classes. In other words, defective solder joints have a higher level of color complexity. Additionally, in terms of correlation feature, normal solder joints show a higher level of pixel dependency compared to defective solder joints. The higher correlation value in normal solder joints indicates a stronger relationship between pixels. Therefore, the correlation feature can be an important factor in distinguishing between the two classes of solder joints.

Table 2. GLCM feature data from 2 class solder joint

 Defected solder joint image	 Normal solder joint image
Feature data: Disimilarity: 2.2700 Energy: 0.7428 Entrophy: 0.6507 Corelation: 0.8334	**Feature data:** Disimilarity: 1.5557 Energy: 0.72678 Entropy: 0.4790 Corelation: 0.9295

Overall, the analysis of texture features in defective and normal solder joints demonstrates significant differences. However, it's important to note that the feature extraction in this study was performed through log-polar transformation, while previous research by [9, 10] conducted texture feature extraction using GLCM directly without log-polar transformation. To determine whether log-polar transform can significantly impact the quality inspection of solder joint smoothness, further testing of texture feature extraction techniques without log-polar transformation is needed.

To investigate this matter, a classification task was conducted with the objective of categorizing solder joint images based on the extracted GLCM texture feature data. The classification process was carried out by applying Eqs. (9-12) and resulted in the identification of two classes: the defect class and the normal class. To evaluate the performance of the classification models, a dataset consisting of 736 solder joint images was prepared, comprising 430 images classified as normal and 306 images representing solder joint defects. The dataset was then processed using the two compared models, yielding detection results as presented in Table 3. However, to comprehensively assess the effectiveness of the models, a crucial step involved

Table 3. Classification result

Result	GLCM without log polar transform	GLCM with log polar transform
TP	366	398
TN	290	302
FP	15	13
FN	65	23

Table 4. Model performance

result	GLCM without log polar transform	GLCM with log polar transform
Accuracy	0.89	0.95
Precision	0.96	0.97
Recall	0.85	0.95
F1-Score	0.90	0.96

evaluating the classification outcomes using a confusion matrix.

In Table 3, TP represents the count of true positives, TN represents the count of true negatives, FP represents the count of false positives, and FN represents the count of false negatives. These values are essential for evaluating the classification performance using a confusion matrix. The accuracy of the classification is calculated as $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, which measures the overall correctness of the classification model. Additionally, precision is computed as $Precision = \frac{TP}{TP+FP}$, representing the model's ability to correctly identify samples as positive. The recall value, calculated as $Recall = \frac{TP}{TP+FN}$, indicates the model's sensitivity or true positive rate (TPR). Finally, the F1-Score is determined as $F1 - Score = \frac{2 \times Precision \times Recall}{Precision+Recall}$, which is the harmonic mean or the balanced average of precision and recall. The evaluation results of the performance of the compared models are presented in Table 4.

Based on the evaluation results obtained from the presented confusion matrix in Table 4, a notable difference is observed between the usage of GLCM with and without log-polar transformation in terms of classification performance.

In the case of utilizing GLCM without log-polar transformation, an accuracy level of 0.89 is attained, reflecting the model's overall classification capability. Precision, which measures the model's accuracy in correctly classifying positive samples, achieves a value of 0.96 in this model. However, the recall, indicating the model's ability to accurately detect positive samples, exhibits a value of 0.85, indicating suboptimal detection of positive samples. The F1-Score, a metric combining precision and recall using harmonic mean, reaches 0.90.

Conversely, when GLCM is employed with log-polar transformation, the accuracy increases to 0.95. Precision demonstrates a value of 0.97, showcasing the model's proficiency in accurately classifying positive samples. Notably, recall attains a higher value of 0.95, indicating an enhanced capability of the model in accurately detecting positive samples with a higher degree of accuracy. The F1-Score also experiences an improvement to 0.96, indicating an overall performance enhancement. The model was subsequently deployed on a Raspberry Pi 4B microcomputer with 4 GB of RAM. Computational time testing revealed an average computational speed of 120 ms. However, it is worth noting that this speed can be further improved through a RAM upgrade to 8 GB.

Based on the evaluation outcomes, it can be inferred that the usage of GLCM with log-polar transformation provides a significant improvement in the detection of positive samples (recall) compared to GLCM without log-polar transformation. Moreover, log-polar transformation contributes to an overall enhancement in accuracy, precision, and F1-Score, demonstrating its effectiveness in bolstering the overall classification performance.

5. Conclusion

This study successfully enhances the computational efficiency for pixel analysis in images of THT solder joints, facilitating the detection of defects. The implementation of log-polar transformation allows for pixel analysis in the polar coordinate system, whereas previously, pixel analysis was performed in the rectangular coordinate system, leading to unnecessary processing of some pixels. The application of GLCM texture feature extraction, which has previously shown good performance in monitoring solder joint quality, becomes more effective and efficient through the addition of the log-polar transformation procedure. Through this enhanced analytical approach, the model achieves a maximum accuracy of 95% with a computational speed of 120 ms. These outcomes align the model's performance with the benchmarks achieved by deep learning methodologies, which typically yield accuracies ranging from 90% to 97.5% with computational times ranging from 390 ms to 100 ms. Although the model has not surpassed the best results of deep learning techniques, it aligns with the initial goal of being deployable on microcomputers, thereby opening possibilities for cost-effective and flexible product development. This research also presents a new opportunity as an open problem in the creation of small and portable quality monitoring tools for THT solder joints, a task that was previously challenging with GPU-based programming.

Conflict of interest

The authors declare no conflict of interest.

Author contribution

The first author was responsible for the methodology, programming, evaluation, investigation, data collection, original draft preparation, and review and editing of the manuscript. The second and third authors were responsible for supervision and project administration.

References

[1] R. Seidel, T. Ahrens, J. Friedrich, A. Reinhardt, and J. Franke, "Experimental identification and prioritization of design and process parameters on hole fill in mini wave soldering", *Microelectronics Reliability*, Vol. 131, No. 114497, 2022.

[2] W. Z. Z. W. Zainudin, T. C. Yong, T. C Hui, Y. B. Kar, and W. Y. Hoong, "Optimization of reflow profile for copper pillar with SAC305 solder cap FCCSP", *Journal of Materials Science: Materials in Electronics*, Vol. 34, No. 3, p. 187, 2023.

[3] J. P. Nayak and B. D. Parameshachari, "Effective PCB Defect Detection Using Stacked Autoencoder with Bi-LSTM Network", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 5, 2022, doi: 10.22266/ijies2022.1031.05.

[4] E. Verna, G. Genta, M, Galetto, and F. Franceschini, "Economic impact of quality inspection in manufacturing: A proposal for a novel cost modeling", *Journal of Engineering Manufacture*, Vol. 236, No. 11, pp. 1508-1517, 2022.

[5] V, Azamfirei, F, Psarommatis, and Y. Lagrosen, "Application of automation for in-line quality inspection, a zero-defect manufacturing approach", *Journal of Manufacturing Systems*, Vol. 67, pp. 1-22, 2023.

[6] A. A. R. M. A. Ebayyeh and A. Mousavi, "A Review and Analysis of Automatic Optical Inspection and Quality Monitoring Methods in Electronics Industry", *IEEE Access*, Vol. 8, pp. 183192-183271, 2020.

[7] J. Zipfel, F, Verworner, M. Fischer, U Wieland, M. Kraus, and P. Zschech, "Anomaly detection for industrial quality assurance: A comparative evaluation of unsupervised deep learning models", *Journal of Computers & Industrial Engineering*, Vol. 177, p. 109045, 2023.

[8] Z. Ren, F. Fang, N, Yan, and Y. Wu, "State of the art in defect detection based on machine vision", *International Journal of Precision Engineering and Manufacturing-Green Technology*, Vol. 9, No. 2, pp. 661-691, 2022.

[9] Y. M. Chang, C. C. Wei, J. Chen, and P. Hsieh, "Classification of Solder Joints via Automatic Mistake Reduction System for Improvement of AOI Inspection", In: *Proc. of 13th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, Taipei, Taiwan, pp. 150-153, 2018.

[10] Y. M. Chang, C. C. Wei, J. Chen, and P. Hsieh, "An Implementation of Health Prediction in SMT Solder Joint via Machine Learning", In: *Proc. of IEEE International Conference on Big Data and Smart Computing (BigComp)*, Kyoto, Japan, pp. 1-4, 2019.

[11] N. Thiele, Z. Jiang, K. Schmidt, R. Seidel, C. Voigt, A. Reinhardt, and J. Franke, "Clustering of Image Data to Enhance Machine Learning Based Quality Control in THT Manufacturing", In: *Proc. of 27th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, Timisoara, Romania, pp. 287-291, 2021.

[12] V. Reshadat and R. A. J. W. Kapteijns, "Improving the Performance of Automated Optical Inspection (AOI) Using Machine Learning Classifiers", In: *Proc. of International Conference on Data and Software Engineering (ICoDSE)*, Bandung, Indonesia, pp. 1-5, 2021.

[13] C. L. S. C. Fonseka and J. A. K. S. Jayasinghe, "Color Model Analysis for Solder Pad Segmentationon Printed Circuit Boards", *International Journal of Scientific and Research Publications*, Vol. 6, No. 11, 2016.

[14] J. D. Song, Y. G. Kim, and T. H. Park, "SMT defect classification by feature extraction region optimization and machine learning", *International Journal of Advanced Manufacturing Technology*, Vol. 101, No. 5–8, pp. 1303–1313, 2019.

[15] G. Mahalingam, K. M. Gay, and K. Ricanek, "PCB-METAL: A PCB Image Dataset for Advanced Computer Vision Machine Learning Component Analysis", In: *Proc. of 16th International Conference on Machine Vision Applications (MVA)*, Tokyo, Japan, pp. 1-5, 2019.

[16] C. L. S. C. Fonseka and J. A. K. S. Jayasinghe, “Implementation of an Automatic Optical Inspection System for Solder Quality Classification of THT Solder Joints”, *IEEE Trans Compon Packaging Manuf Technol*, Vol. 9, No. 2, pp. 353–366, 2019.

[17] W. Zhao, S. R. Gurudu, S. Taheri, S. Ghosh, M. A. M. Sathiaseelan, and N. Asadizanjani, “Pcb component detection using computer vision for hardware assurance”, *Journal of Big Data and Cognitive Computing*, Vol. 6, No. 2, p. 39, 2022.

[18] M. B. Kiran, “Vision-Based Printed Circuit Board Inspection”, In: *Proc of the 7th North American International Conference on Industrial Engineering and Operations Management*, Orlando, Florida, USA, 2022.

[19] W. Tangsuksant, J. Inthiam, H. Silangren, P. Iamcharoen, and W. Jitviriya, “Defect Solder Classification in Print Circuit Boards using Machine Learning”, In: *Proc. of International Conference on Artificial Life and Robotics (ICAROB2023)*, Oita, Japan, 2023.

[20] W. Li, B. Esders, and M. Breier, “SMD segmentation for automated PCB recycling”, In: *Proc. of 11th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 65-70, 2013.

[21] W. Dai, A. Mujeeb, M. Erdt, and A. Sourin, “Soldering defect detection in automatic optical inspection”, *Journal of Advanced Engineering Informatics*, Vol. 43, p. 101004, 2020.

[22] S. Q. Ye, C. S. Xue, C. Y. Jian, Y. Z. Chen, J. J. Gung, and C. Y. Lin, “A Deep Learning-based Generic Solder Defect Detection System”, In: *Proc. of International Conference on Consumer Electronics-Taiwan*, pp. 99-100, 2022.

[23] Y. M. Chang, T. L. Lin, H. C. Chi, and W. K. Lin, “Deep Learning-based AOI System for Detecting Component Marks”, In: *Proc. of IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 243-247, Jeju, Korea, pp. 243-247, 2023.

[24] I. C. Chen, R. C. Hwang, and H. C. Huang, "PCB Defect Detection Based on Deep Learning Algorithm", *Journal of Processes*, Vol. 11, No. 3, p. 775, 2023.

[25] M. Fettke, G. Bonow, A. Fisch, M Nasser, G. Friedrich, R. Baba, and T. Teutsch, “Forming of advanced THT-interconnects using SB2 laser solder jetting process”, In: *Proc. of 73rd Electronic Components and Technology Conference (ECTC)*, pp. 643-650, IEEE, 2023.

[26] N. Sabri, M. Mukim, Z. Ibrahim, N. Hasan, and S. Ibrahim, "Computer Motherboard Component Recognition Using Texture and Shape Features", In: *Proc. of 9th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, Shah Alam, Malaysia, pp. 121-125, 2018.

[27] T. Li, A. He, L. Kong, M Zhang, C. Liu, X. Ma, and Y. Xu. "Application of Machine Learning Algorithms in Cold-Rolled Strip Steel Surface Defect Grading Systems", In: *Proc. of International Conference on Automation and Computing (ICAC)*, Birmingham, United Kingdom, pp. 1-6, 2023.

[28] R. M. Haralick, K, Shanmugam, and I. H. Dinstein, “Textural features for image classification”, *Journal of Transactions on Systems, Man, and Cybernetics*, Vol. SMC-3, No. 6, pp. 610-621. 1973.

[29] A. Septiarini, R. Saputra, A. Tedjawati, M. Wati, and H. Hamdani, "Pattern Recognition of Sarong Fabric Using Machine Learning Approach Based on Computer Vision for Cultural Preservation", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 5, 2022, doi: 10.22266/ijies2022.1031.26.

[30] I. E. Araar, A. Benammar, R. Drai, and A. Benyahia, "A GLCM-Based Approach for the Clustering of Weld Joint Images", In: *Proc. of Fifth International Conference on Intelligent Computing in Data Sciences (ICDS)*, Fez, Morocco, pp. 1-9, 2021.

[31] G. Singh, N. Das, R. K. Panda, B. P. Mohanty, D. Entekhabi, and B. K. Bhattacharya, “Soil moisture retrieval using SMAP L-band radiometer and RISAT-1 C-band SAR data in the paddy dominated tropical region of India”, In: *Proc. of IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 14, pp. 10644-10664, 2021.

[32] R. Patel, C. I. Patel, and A. Thakkar, "Aggregate features approach for texture analysis", *Nirma University International Conference on Engineering (NUiCONE)*, Ahmedabad, India, pp. 1-5, 2012.

[33] N. Goel and D. Ganotra, "An Approach for Anti-Contrast Enhancement Forensic Detection Based on Glcm and Zernike Moments", *International Journal of Information Technology*, Vol. 15, pp. 1625–1636, 2023.

[34] H. Wang, S. Li, H. Qiu, Z. Lu, Y. Wei, Z. Zhu, and H. Ge, “Development of a Fast Convergence Gray-Level Co-Occurrence Matrix for Sea Surface Wind Direction Extraction from Marine Radar Images”, *Journal of Remote Sensing*, Vol. 11, No. 8, p. 915, 2023.

[35] S. Zokai and G. Wolberg, “Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations”,

Journal of IEEE Transactions on Image Processing, Vol. 14, No. 10, pp. 1422-1434, 2005.

[36] B. Mazaheri and H. Pourghassem, "Iris image classification based on texture and Fourier Mellin Transform features", In: *Proc. of IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, China, pp. 118-122, 2011.

[37] M. A. Taha, M. A. Hanaa, and O. H. Saif, "Iris Features Extraction and Recognition based on the Scale Invariant Feature Transform (SIFT)", *Webology*, Vol. 19, No. 1, pp. 171-184, 2022.