METAL SURFACE IRREGULARITY DETECTION

A report on
Metal surface irregularity detection using GLCM
[CSE-3181]

Submitted By
Joy Podder 210962184
Anurag Kasat 210962180
Vishal Agarwal 210962148

MANIPAL ACADEMY of HIGHER EDUCATION
(Institution of Eminence Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MANIPAL INSTITUTE OF TECHNOLOGY,
MANIPAL ACADEMY OF HIGHER EDUCATION

NOVEMBER 2025

Metal surface irregularity detection by computing GLCM

Joy Podder[1], Anurag Kasat[2], Vishal Agarwal[3]

[1]CSE AIML MIT Manipal, India

[2]CSE AIML MIT Manipal, India

[3]CSE AIML MIT Manipal, India

[1]jpodder77@gmail.com; [2]anuragkasat1606@gmail.com; [3]agarwal.vishal2003@gmail.com

I. INTRODUCTION

Metal surface irregularities can lead to a range of problems across various industries. These include reduced structural integrity, corrosion, machinery inefficiencies, leaks, electrical conductivity issues, aesthetic concerns, safety hazards, dimensional inaccuracies, increased wear and tear, quality assurance issues, and regulatory non-compliance. To mitigate these issues, it's essential to implement rigorous quality control measures, including regular inspections, testing, and appropriate surface treatment processes. Advanced technologies like non-destructive testing and automated inspection systems can be valuable in detecting and addressing irregularities before they escalate into significant problems.

One promising approach to address surface irregularities in metals involves the use of Gray-Level Co-occurrence Matrix (GLCM) based feature extraction. This technique is a powerful tool in the field of computer vision and image processing. It is particularly well-suited for texture analysis in images, making it an excellent choice for metal surface inspection. The GLCM is a matrix that quantifies the co-occurrence of pixel values at different spatial relationships within an image. By calculating the relative frequency of each pair of gray levels, the GLCM encapsulates crucial information about the texture of the surface. Once the GLCM is computed, a range of statistical measures can be extracted to characterize the texture. These features include contrast, homogeneity, energy (angular second moment), entropy, correlation, and dissimilarity.

II. LITERATURE REVIEW

Currently, the inspection of metal surfaces relies heavily on visual inspection by human operators or conventional image processing techniques, which are often limited in their ability to detect subtle irregularities. Machine learning-based approaches are as follows:

A. Real-Time Metal-Surface irregularity detection by calculating Renyi's entropy

As seen in [1] the challenge of detecting and classifying metal surface defects in real time using video streams has been tackled using Renyi's entropy. In order to extract important information and features from video streams the model uses Renyi's entropy. Its effectiveness can be seen through the results obtained, when they are compared with results from decision tree classifier it can be seen that the model using Renyi's entropy outperforms conventional models.

B. Deep Learning Approach for Metal Surface Defect Detection

As seen in [2] the authors use the TLU framework in order to detect surface irregularity . 2 types of encoders ResNet and DenseNet have been used. The performance of these nets has been compared using random initialization and the ImageNet dataset has been used for training purposes. The results show that the TLUnet performs better than random initialization in classification by 5% and segmentation by 26%.

C. Automatic Metallic Surface Defect Detection and Recognition with a Novel Cascaded Autoencoder (CASAE)

The paper [3] presents a two-step method for automatically detecting and identifying metallic defects from real-world images. The method uses a new cascaded autoencoder (CASAE) model to segment and locate the defects. The CASAE model produces a pixel-level prediction mask based on semantic segmentation. The defect areas in the segmented images are then classified into different categories using a CNN network.

D. Detection of irregularities using IR laser diode

In this [4] study, the authors Zhu, X., Liu, J., Zhou propose a method based on deep learning for detecting metal surface irregularities. First, they create their dataset. Next, they design a new feature extraction layer based on YOLOv5s for detecting irregularities. Further, an attention mechanism is introduced into the model to help extract features of defects. The improved model is trained and tested on the dataset. The test results show that the accuracy of the improved model is 3.8% better than the original detection model. When compared with other irregularity detection techniques, the accuracy of the improved model achieves very good results.

III. METHODOLOGY

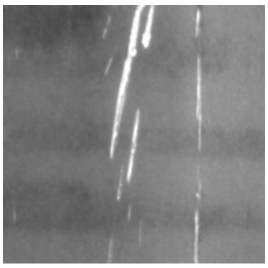
A. Image processing

1. To smoothen the image and facilitate edge detection we have initially used a Gaussian Filter, it does so by reducing noise in the image, kernel size used is 5*5.

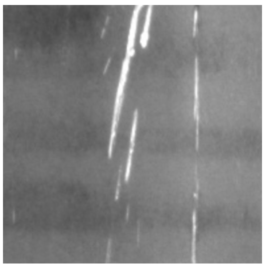
$$G_0(x,y) = Ae^{\frac{-(x-\mu_x)^2}{2\sigma_x^2}+\frac{-(y-\mu_y)^2}{2\sigma_y^2}}$$

2D gaussian is shown by the above equation, where μ is the mean and σ^2 is the variance.

2. Next, we use canny edge detection to highlight and visualize the edges in the image after the Gaussian blur has been applied. The lower threshold has been set to 10 and the upper threshold has been set to 130.
   Canny edge involves computing the edge gradient of the image using the horizontal and vertical derivatives obtained from the Sobel kernel. Then non-max suppression is carried out to obtain a binary image with thin edges.
   Finally, it uses hysteresis thresholding to determine strong edges based on the intensity gradient thresholds.



Original image



Gaussian Blurred image

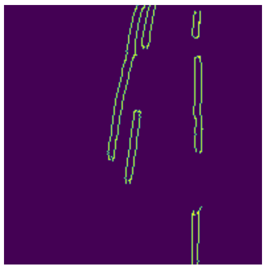


Image after canny edge detection

B. GLCM calculation

1. GLCM has been used to generate and extract features which would ultimately help in classification of metal surfaces. The GLCM is a matrix that quantifies the co-occurrence of pixel values at different spatial relationships within an image. By calculating the relative frequency of each pair of grey levels, the GLCM encapsulates crucial information about the texture of the surface.
2. The computation of co occurrence matrix of GLCM depends on 3 parameters:
   i. Pixel distance
   ii.Position angle
   iii.Gray levels
3. We have chosen the pixel distance as 4 and chosen 6 position angles 0, $\pi/4$, $\pi/3$, 2 $\pi/3$, π, 4 $\pi/3$ radians and number of gray levels as 256. The grey level parameter influences the size of the GLCM.

4. The co occurrence matrix can be created with the following formula:

$$C_{ij} = \frac{\delta_{ij}}{\sum_{i=0}^{N} \sum_{j=0}^{N} \delta_{ij}}$$

Here δ_{ij} represents the number of gray level occurences within the given window and C_{ij} represents the co-occurrence matrix after normalization.

Sample GLCM calculation for an arbitrary matrix where distance =1 and position angle = 0

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

Initial image

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

GLCM matrix of the above image

From the GLCM matrix we can see that element (7,1) contains the number 2, this is because there are 2 instances in the image where 7 and 1 are adjacent to each other. Similarly, the zeros indicate that no pixels having intensities corresponding to the index of the GLCM matrix are horizontally adjacent to each other.

C. Feature selection

To compute the similarity between different Grey Level Co-Occurrence matrix, we have used 4 features contrast, correlation, homogeneity and dissimilarity.

Out of these contrast, correlation and homogeneity are chosen keeping results from [5] in mind.

1. Contrast:
   The contrast feature in the gray-level co-occurrence matrix (GLCM) measures the local intensity variation in an image. It represents the difference between the highest and lowest values of adjacent pixel pairs in the GLCM. Highly textured images have high contrast values whereas smooth images have lower contrast values.
   Contrast can be calculated by the following equation:

$$C = \sum_{i=0}^{j-1} N^2 \left\{ \sum_{|i-j|=0}^{N-1} j = 1 \sum_{j=1}^{N-1} C_{ij} \right\}$$

2. Correlation:
   Also known as entropy, this measures the linear dependency between the grey levels of adjacent pixel pairs in an image. It is a measure of the strength of linear relationships between neighbouring pixels. A higher correlation value would indicate a stronger linear relationship or a uniform image. Whereas a weaker linear relationship would indicate irregular texture pattern.
   Correlation can be calculated by this formula:

$$E = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} i, jC_{ij} - \mu_x(C_{ij})\mu_y(C_{ij})}{\sigma_x(C_{ij})\sigma_y(C_{ij})}$$

   Here, mean of the image is computed as follows:

$$\mu_x = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} iC_{ij}$$

$$\mu_y = \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} jC_{ij}$$

   Standard deviation is given by the following expression:

$$\sigma_x = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (1 - \mu_x)^2 C_{ij}$$

$$\sigma_y = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (1 - \mu_y)^2 C_{ij}$$

3. Homogenity:
   It is a measure of the uniformity or smoothness of the texture of an image. It indicates how likely neighbors are to to have the same gray level values. An image with low homogenity values suggest a more heterogeneous and diverse image.
   It is calculated by the following formula:

$$H = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{C_{ij}}{1 + |i - j|}$$

4. Dissimilarity:
   Dissimilarity is a measure of local intensity variation in an image it measures the average difference in gray levels between adjacent pixel pairs in an image. It indicates the overall variation in intensity levels within the image. A higher dissimilarity value suggests a greater difference in intensity levels between neighboring pixels, suggesting a more textured image. Similarly, a lower dissimilarity value implies a smaller difference in intensity levels, suggesting a more uniform image.

The dissimilarity can be calculated using the following formula:

$$\sum_{i,j=0}^{N-1} C_{ij} \left|i - j\right|$$

D. Classification based on features

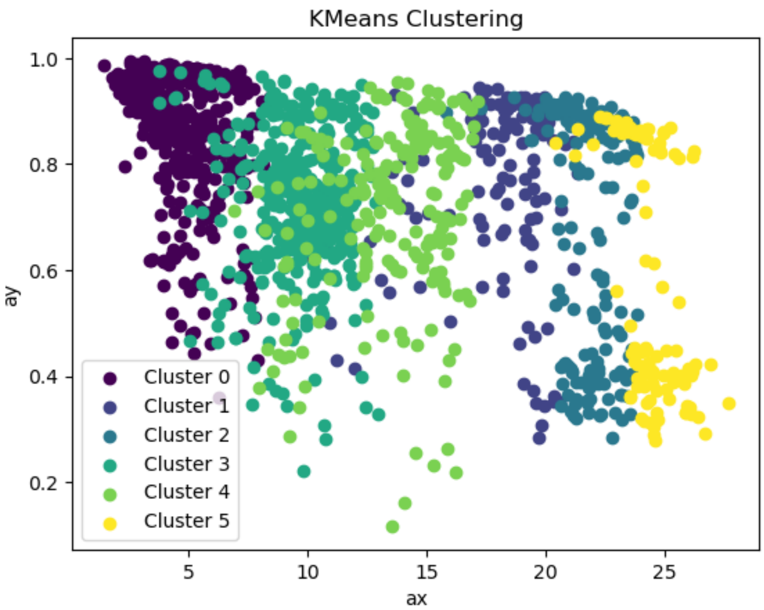
1. Features extracted from the GLCM matrix are stored in a dataframe and passed through various classification algorithms, The algorithms used were:
   i. Kmeans Clustering
   ii.SVM classifier
   iii.Decision Tree classifier

2. Kmeans Clustering:
   This is an unsupervised ML algorithm which groups data into clusters, here K defines the number of predefined clusters. Each cluster has a centroid associated with it and the algorithm classifies points by minimizing the distance between input data points and cluster centres.

   In our case K has been set to 6 to classify the metal surface into one of the 6 irregularity classes we have taken. Euclidean distance is the distance measure used to measure the distance between datapoints and centroids. It is represented by the following formula.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

   Initially we assume centroids randomly and calculate the distance of each datapoint from the centroids, we assign a datapoint to the cluster to which it is the closest, based on these assignments we recalculate the centroids as the mean of the datapoints. We keep repeating this process until the coordinates of the centroids converge.
   The clusters formed are as shown below:



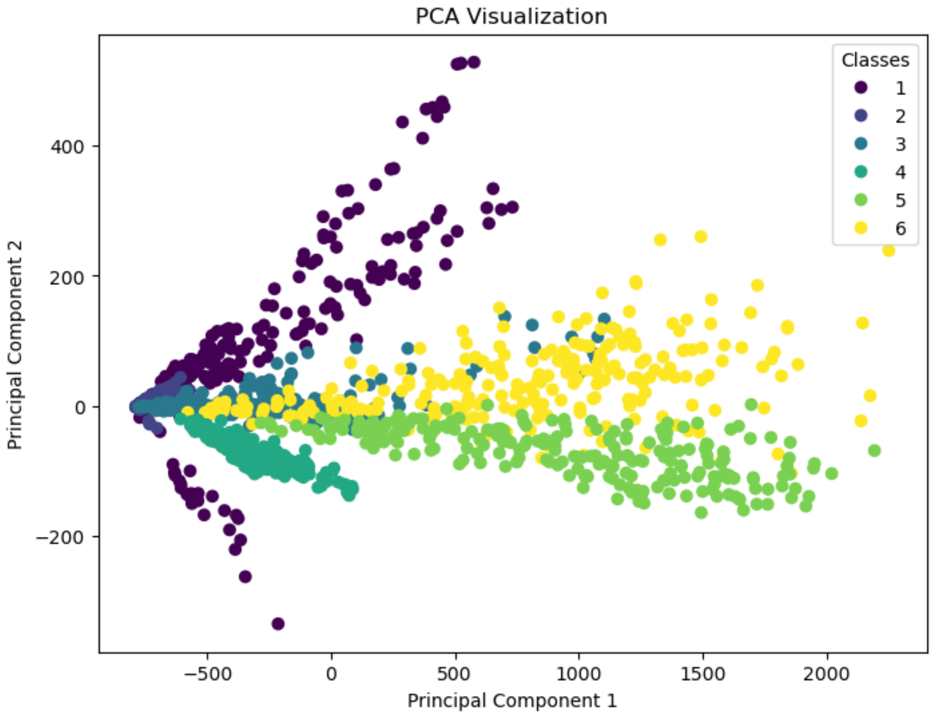
3. SVM Classifier:
   SVM is a supervised ML classification algorithm, it helps to classify datapoints by finding an optimal hyperplane in the n dimensional space. An optimal hyperplane is one which maximizes the margin (distance between nearest points belonging to different classes).

The dimension of a hyperplane defines upon the number of features present. If the number of features is p the dimension of the hyperplane is p-1.

In our case the data is non-linear hence a Gaussian rbf kernel is to be used which increases dimensionality of the datapoints and makes them easier to separate. Its expression is given by:

$$k(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right)$$

The clusters formed from this model is as shown below:



We have chosen this seeing Nhat-Duc Hoang's, Van-Duc Tran's results from [6].

4. Decision tree classifier:
   Decision trees consist of root nodes, branches and leaf nodes. This algorithm helps to classify a datapoint into classes by building a tree like structure where internal nodes are decided by applying thresholds on certain attributes of the dataset. Each branch represents a specific range in which the value of a feature belongs. Branches are formed when the best attribute to split the data on is found.

   In order to select the best attribute the following entropy function is used

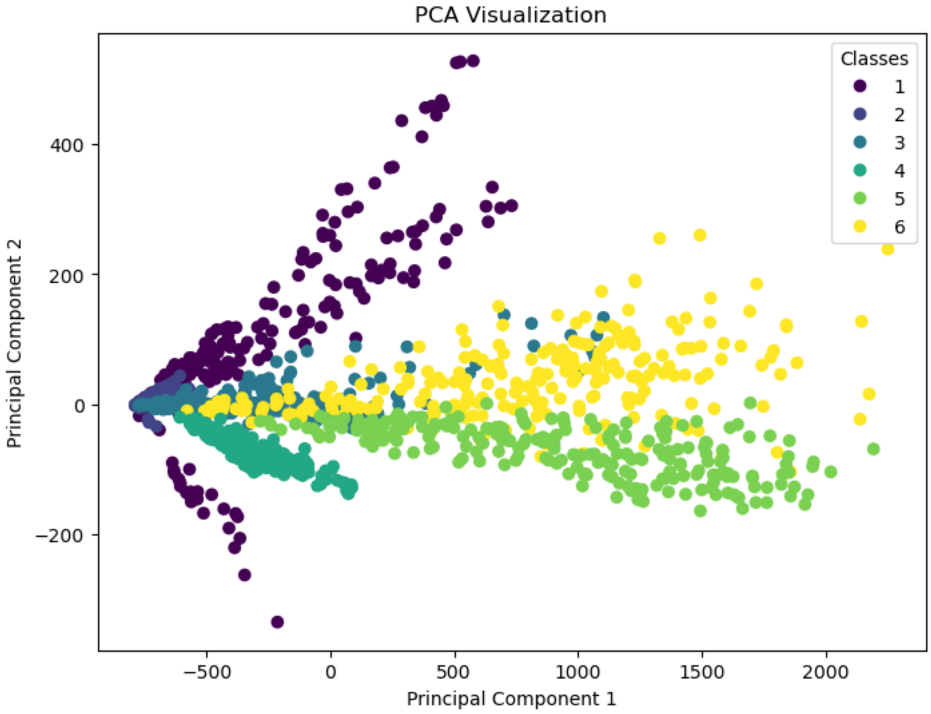
$$Entropy = \sum_{i=1}^{C} - p_i * \log_2(p_i)$$

   Here p_i is the proportion of class i in the dataset. From the value of entropy obtained we can calculate the information gain value of an attribute.

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in D_A} \frac{|S_v|}{|S|} Entropy(S_v)$$

   Attribute with the highest information gain value is chosen to carry out splitting.

The results obtained from the above model is as follows:



IV. Experimental setup

In order to carry out this project we have used the NEU-DET dataset which consists of 6 types of surface defects namely

1. Rolled-in-scale
2. Patches
3. Crazing
4. Pitted surface
5. Inclusion
6. Scratches

We have used a total of 1440 pictures from the dataset.

Furthermore, we have calculated and selected 4 features namely contrast, correlation, dissimilarity, homogeneity from the Gray level co-occurrence matrix of these images along 6 distinct angles leading to a total of 24 features to train the model.

V. Results and discussion

A. Performance measures

1. Accuracy: Accuracy is the proportion of correctly classified instances. It is defined as the sum of true positives (TP) and the true negatives (TN) divided by sum of number of instances (TP + TN + FP + FN).

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

2. Precision: It is the proportion of predicted positive instances that are positive. It is calculated as the number of true positives (TP) divided by the sum of predicted positives (TP + FP).

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

3. Recall: Recall, also known as sensitivity, is the proportion of instances that are actually positive and correctly classified. It is calculated as true positives (TP) divided by actual positives (TP + FN).

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

4. F1 score: This is a HM of precision and recall, which means it balances both metrics equally. It is calculated as the following:

$$\text{F1 score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

5. Support: Support is the total number of instances in a particular class. It is calculated as the sum of all the instances present for that class.

$$\text{Support} = \text{TP} + \text{FP} + \text{FN} + \text{TN}$$

B. Classification results

1. Decision Tree Classifier:

```
Accuracy: 0.8645833333333334
Classification Report:
              precision    recall  f1-score   support

           1       0.83      0.93      0.88        42
           2       0.91      1.00      0.95        49
           3       0.76      0.60      0.67        52
           4       0.92      0.94      0.93        48
           5       0.95      0.87      0.91        45
           6       0.82      0.88      0.85        52

    accuracy                           0.86       288
   macro avg       0.86      0.87      0.86       288
weighted avg       0.86      0.86      0.86       288
```

2. SVM classifier:

```
Accuracy: 0.9097222222222222
Classification Report:
              precision    recall  f1-score   support

           1       1.00      0.86      0.92        42
           2       0.82      1.00      0.90        49
           3       0.85      0.85      0.85        52
           4       0.94      1.00      0.97        48
           5       0.94      1.00      0.97        45
           6       0.98      0.77      0.86        52

    accuracy                           0.91       288
   macro avg       0.92      0.91      0.91       288
weighted avg       0.92      0.91      0.91       288
```

3. KMeans classifier:
Since this is an unsupervised algorithm it has different performance measures compared to the other two
Silhouette Score: This ranges from -1 to 1, a higher value indicates a datapoint is matched well to its own cluster and poorly matched to neighboring clusters.
Silhouette score value: 0.5234638369574687

VI. Conclusions

Defects to metals cause damage to property and human resources every year. These defects need to be detected and corrected as soon as possible to mitigate any calamity. It is not efficient or affordable to implement manual inspection at such a large scale. After extracting 24 features for 1440 metal images our machine learning model has a proposed accuracy of 86% with the decision tree classifier and 90% with the SVM classifier. Therefore, this model can serve as a useful tool for efficiently detecting metal surface irregularities.

VII. FUTURE WORK

Our work can be carried forward further by the utilization of deep-learning and advanced ML techniques and algorithms in order to predict the exact pixel coordinates of the deformities. We can also implement depth analysis and size estimation techniques to determine the extent to which damage has been done to the surface of the metals.

REFERENCES

[1] W. Liu, K. Yan, H. -C. Wu, X. Zhang, S. Y. Chang and Y. Wu, "Real-Time Metal-Surface-Defect Detection and Classification Using Advanced Machine Learning Technique," 2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Bilbao, Spain, 2022, pp. 1-5, doi: 10.1109/BMSB55706.2022.9828748..

[2] TLU-Net: A Deep Learning Approach for Automatic Steel Surface Defect Detection Praveen Damacharla, Achuth Rao M. V., Jordan Ringenberg, Ahmad Y JavaidS. arXiv:2101.06915

[3] Tao, Xian, Dapeng Zhang, Wenzhi Ma, Xilong Liu, and De Xu. 2018. "Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks" Applied Sciences 8, no. 9: 1575.

[4] Zhu, X., Liu, J., Zhou, X. et al. Detection of irregular small defects on metal base surface of infrared laser diode based on deep learning. Multimed Tools Appl (2023). https://doi.org/10.1007/s11042-023-16352-3.

[5] Deotale, N.T., Sarode, T.K. Fabric Defect Detection Adopting Combined GLCM, Gabor Wavelet Features and Random Decision Forest. 3D Res 10, 5 (2019).

[6] Nhat-Duc Hoang, Van-Duc Tran, "Image Processing-Based Detection of Pipe Corrosion Using Texture Analysis and Metaheuristic-Optimized Machine Learning Approach", Computational Intelligence and Neuroscience, vol. 2019, Article ID 8097213.