https://jurnaleeccis.ub.ac.id/
e-ISSN(Online): 2460-8122

Jurnal EECCIS Vol. 18, No. 1, April 2024
pp 19-22

Marble Surface Anomaly Detection using Autoencoder Architecture

Muhammad Yahya Abdullah [1], Cemil Öz [2]
[1,2] Faculty of Computer Engineering, Sakarya University, Sakarya, Türkiye
Email: yahya.abdullah@ogr.sakarya.edu.tr, coz@sakarya.edu.tr

***Abstract*— Marble is a material that is commonly used for building components such as furniture, flooring, countertops, bathrooms, windows in homes etc. Due to the many uses of marble in various aspects, marble surface detection is important for this industry to improve quality and avoid financial problems that may occur. In previous research, many methods such as wavelet transform, Gabor transform, co-occurrence matrix and artificial neural network were implemented in defect detection (fabric or other tasks). In this study we built a platform that aims to detect anomalies on marble surfaces using Autoencoders architecture, Keras library and Python programming language. To test the model that has been created, a marble surface dataset obtained from kaggle.com, one of the largest dataset provider sites, was used and an accuracy of 89% was obtained. The conclusions of this study include the effectiveness of this method in detecting anomalies, the advantages of the autoencoder architecture compared to other methods, and the potential practical applications of these findings in various fields. By utilizing the autoencoder's ability to reconstruct data, anomaly detection can be performed by comparing the reconstructed results with the original data. The main advantage of this approach lies in its ability to tackle the problem of anomaly detection without the need for class labels.**

***Index Terms*—Anomaly Detection, Autoencoder Architecture, Marble Surface.**

I. INTRODUCTION

Marble is a versatile and durable material, its use in various applications and industries continues every year, the number of marble users will also continue to fluctuate based on trends and economic conditions in various regions, as well as demand related to special projects. Since many consumers buy marble because of its beautiful patterns and long durability, it is very important for marble factories to improve the monitoring of marble surface faults so as to maintain consumer confidence to continue buying marble at the factory. In addition, early prevention of problems caused by faulty production prevents material losses that might otherwise occur [1, 2]. For these reasons, it is crucial to detect surface faults in the manufacturing industry.

Conventional surface defect inspection is done manually under human supervision [3, 4]. This method has disadvantages such as wasted time, low efficiency, and subjective results as it depends on many factors such as employee experience and motivation [5]. Therefore, we need to use evolving technologies for practical solutions in manufacturing systems. Automated surface defect inspection, using Currently, computer vision, image processing, and machine learning technologies have been applied to automated surface defect detection systems and successful results have been obtained [4], [6].

The main challenges faced by anomaly detection systems are as follows. Anomalies such as dents, stains, cracks, scratches, dirt, scratches, etc., in addition to this, data also often contains noise, thus changing the data to be abnormal and it is not uncommon for the system to classify it as an anomaly. Environmental factors such as lighting, temperature, extreme weather (such as snow) also impact the detection system [7, 8]. These challenges make the task of anomaly detection on textured surfaces complex and difficult.

In some previous studies in surface defect detection, several methods have been used such as Result Weighting-based Resnet Feature Pyramid Network (SA-ROPA) conducted by Hüseyin Üzen and his two friends Muammer Türkoğlu and Davut Hanbay with the article title "Resnet Attribute Pyramid Network Architecture for Surface Error Detection" [9], Convolutional Neural Network (CNN) conducted by Benjamin Staar, Michael Lütjena and Michael Freitaga with the article title "Anomaly detection with convolutional neural networks for industrial surface inspection" [10], Generative Adversial Network (GAN) conducted by Oliver Rippel, Maximilian Muller and Dorit Merhof with the article title GAN-based Defect Synthesis for Anomaly Detection in Fabrics [11], Segmentation-based deep learning conducted by Domen Tabernik, Samo Šela, Jure Skvarc and Danijel Skocaj with the article title "Segmentation-based deep-learning approach for surface-defect detection" [12].

In this research we built a platform that aims to detect anomalies on marble surfaces using the Autoencoders architecture. This platform can be used for factories that produce marble, for the purpose of improving the quality of checking marble surfaces. If there is a company or a person who wants to test their marble, then they simply take a photo of the surface of the marble they want to test then the photo is entered into the system after which the system will process whether there are anomalies or defects such as (cracks, points, joints, etc.) on the marble surface, then the system will give the results in the form of images and categorize the marble into good or bad categories

II. AUTOENCODER ALGORITHM

Autoencoder is one of the machine learning technologies that can be used to reduce data complexity through a computational process involving a neural network (NN), Neural network is the basic concept of deep learning which consists of many neurons that are interconnected and work together to solve certain problems. Autoencoder works by compressing the input data into a compressed representation and then reconstructing the input data from the compressed representation. Autoencoder has two main parts or components, namely the encoder and decoder [13]. The encoder takes the input data and converts it into a smaller representation (encoding), while the decoder converts the representation back into data that is similar to the original (decoding). In this way, the autoencoder can learn to represent data more efficiently.

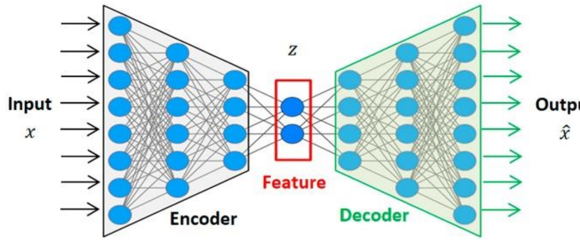

Fig. 1. Flow autoencoder.

Autoencoders can be used for various purposes, such as data compression, dimensionality reduction, or anomaly detection. However, like other machine learning technologies, autoencoder also requires sufficient training data to work effectively.

In this research we use the Autoencoder algorithm, as explained above there are 2 main components that we make, namely the encoder and decoder. In the encoder component, we create 3 layers to compress data into 16 x 16 x 3 with input data image of 128 x 128 x 3. Then in the encoder component we also create 3 layers with compressed data input in the encoder component.

III. EXPERIMENTAL STUDIES

To test the architecture that has been created we use a dataset that we get from kaggle.com in the form of marble surface images. Our goal in this research is to detect anomalies on marble surfaces through images using Autoencoder architecture.

A. Dataset

The dataset we use has two classes train and test. Within the train and test classes there are 4 classes namely: crack, dot, good, and joint. There are a total of 2249 files in the train class with details of 984 files in the crack class, 92 files in the dot class, 860 in the good class and 313 files in the joint class, and a total of 688 files in the test class with details of 246 files in the crack class, 24 files in the dot class, 340 files in the good class and 78 files in the joint class. The images have a dimension of 256 x 256. Due to lack of availability of crack, dot and joint class images, they were augmented using a script. These augmentations were of no loss type, only a change in brightness and an inversion (both horizontal and vertical) of the image was performed. This dataset was created by Vibhor Deshmukh as derector technology at Xoriant. This dataset was published on the kaggle.com website in 2021 [14].

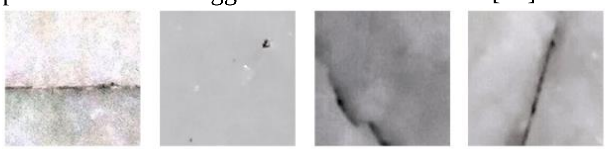

Fig. 2. Sample anomaly (crack, dot and joint) images.

Fig. 3. Sample good images.

B. Implementation Details

To train the model that we have created we use 30 epoch counts, 50 batch sizes, adam optimizer and 'mean_squared_error' as a loss function. all of these experimental studies were conducted on a laptop with system specifications intel processor i5-8250U, 16 GB RAM, 9 GB GPU NVIDIA GeForce MX130. To create and display the matrix table of the predicted results we used the seaborn module and confusion matrix as shown in Figure 2, then to calculate the accuracy in this study we used the accuracy score module from scikit-learn with the formula as shown in Equation 1. In addition to detecting anomalies in the tested images we created a parameter in the form of 'reconstruction error threshold'. With the parameters that have been made, we classify the input image data into anomalous data or good data.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

TABLE I
CONFUSION MATRIX TABLE

True Label	**True Positive**	**False Positive**
	False Negative	True Negative
	Prediction Label	

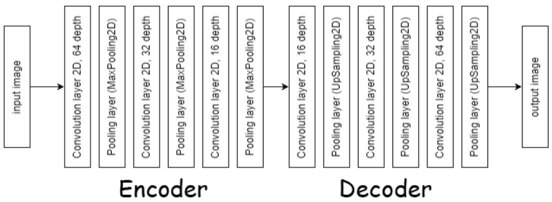

Fig. 4. Purposed model

The image above is the autoencoder architecture that we use where there are several convolution layers and pooling layers. In the encoder section we create a convolution layer from 64 to 16 pixels in size and the pooling layer uses MaxPooling, while in the decoder section the convolution layer is from 16 to 64 pixels in size and the pooling layer uses UpSampling. Previously we have made the convolution layer up to 8 pixels but got bad results, this is because too little data is taken or too much important data/information is wasted. To train the model, we only use good image data (not anomalous images) to get the threshold that will be used as a parameter in the next section.

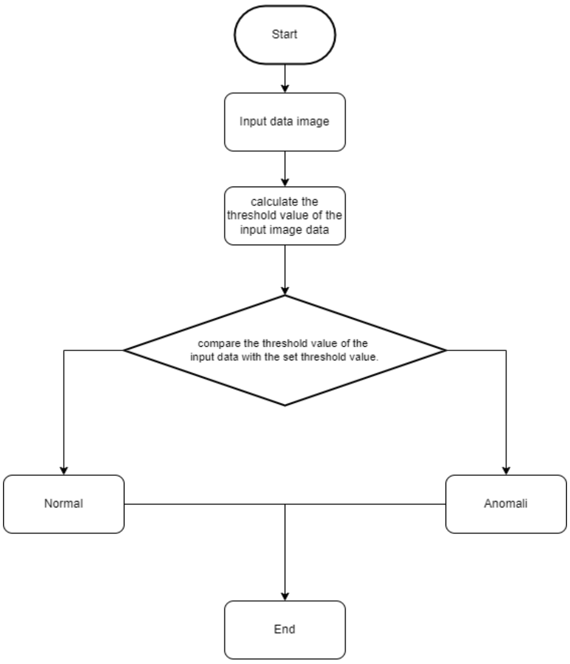

Fig. 5. Flowchart Anomaly Detection.

After the model is trained and the threshold value is obtained, we then create an algorithm to detect anomalies in the input data as shown in the flowchart image above. First we need to enter the input image data to be detected, then the data will be trained to obtain the threshold value using the previously trained model, then after the value has been obtained, the value will be compared with the predetermined threshold value. If the input data value is smaller than the predetermined threshold value then the data will be considered as good data (not anomalous), but if the input data value is greater than the predetermined threshold value then the data will be considered as anomalous data.

IV. Result

We show the success rate using confusion matrix [15, 16]. Confusion matrix shows the quality of learning, rows show the predicted class while columns show the actual class of the sample. The bottom right column shows the number of correctly predicted anomaly images, the bottom left column shows the number of incorrectly predicted anomaly images. The top right column shows the number of good images predicted incorrectly while the top left column shows the number of good images predicted correctly as shown in table 1.

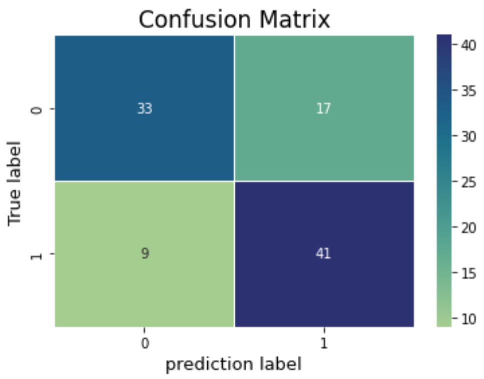

Fig. 6. First trial result on confusion matrix.

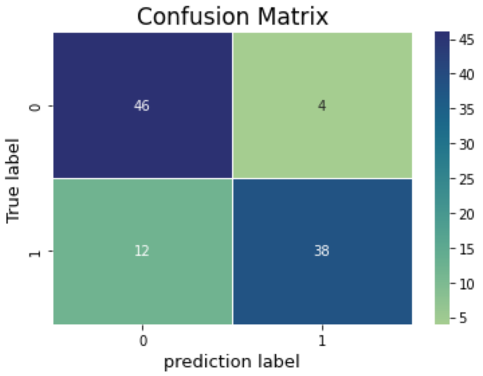

Fig. 7. Second trial result on confusion matrix.

In the first test we used 300 image data trained on the autoencoder model to get the threshold reconstruction error as a parameter for detecting anomalies. Then to test the algorithm that has been made we prepare 50 anomaly images and 50 good images and get a success rate of 74% with a total of 74 correct data and a total of 26 incorrect data as shown in figure 6. Then in the second experiment we increased the amount of image data trained to 400 images with the aim of getting a better threshold reconstruction error. After testing with the same test data as before, namely 50 anomaly images and 50 good images, we obtained an accuracy of 84% with a total of 84 correct data and a total of 16 incorrect data as shown in Figure 7.

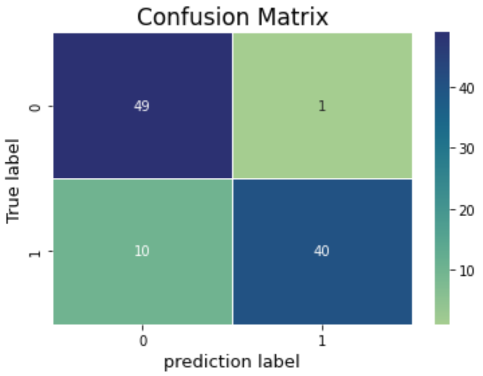

Fig. 8. Third trial result on confusion matrix.

Then in the third experiment we trained the model that had been made using only 860 good image data, 80% for training data and 20% made for validation data. After the model is trained we take the threshold value of the loss function value which will be used as a parameter to detect anomalies. Then to detect anomalies just like the previous experiment we used 50 good image data and 50 anomalous image data. After testing, an accuracy result of 89% was obtained.

In addition to the accuracy value, precision, recall and f1 values were also obtained from the trials. After 3 trials with two different models and three different thresholds, it was seen that the third trial gave the highest results with 89% accuracy as shown in table 2 below.

TABLE II
PREDICTION RESULT TABLE

Dataset	Trial	Accuracy	Precision Score	Recall Score	F1 Score
Marble Surface Anomaly Detection	1	74%	88%	70%	80%
Marble Surface Anomaly Detection	2	84%	90%	76%	83%
Marble Surface Anomaly Detection	3	89%	97%	80%	88%

V. CONCLUSION AND SUGGESTION

In this research we apply one of the deep learning models, Autoencoder, to create an anomaly detection algorithm. In this research, finding the threshold is very important as a parameter used to detect anomalies. Therefore, it is necessary to conduct several tests to determine the number of layers and several hyper parameters (number of iterations, optimizer, input size, and batch size) that are appropriate. As explained above we created 3 layers in the encoder component and 3 layers also in the decoder component, for the input image size we have fixed 128 pixels. And to test the models and algorithms that we have made we use the dataset "marble surface anomaly detection". The results of our research obtained an accuracy value of 89%. The conclusions of this study include the effectiveness of this method in detecting anomalies, the advantages of the autoencoder architecture compared to other methods, and the potential practical applications of these findings in various fields. By utilizing the ability of the autoencoder to reconstruct data, anomaly detection can be performed by comparing the reconstructed results with the original data. The main advantage of this approach lies in its ability to address the problem of anomaly detection without the need for class labels. In future research, improvements can be made by adding explanations of anomaly details (cracks, joints, points, etc.) to the anomaly detection results.

REFERENCES

[1] H. Uzen, M. Turkoglu, and D. Hanbay, “Texture defect classification with multiple pooling and filter ensemble based on deep neural network,” *Expert Systems with Applications*, vol. 175, p. 114838, Aug. 2021, doi: 10.1016/j.eswa.2021.114838.

[2] D. Zhang, K. Song, J. Xu, Y. He, M. Niu, and Y. Yan, “MCnet: Multiple Context Information Segmentation Network of No-Service Rail Surface Defects,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021, doi: 10.1109/TIM.2020.3040890.

[3] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, “PGA-Net: Pyramid Feature Fusion and Global Context Attention Network for Automated Surface Defect Detection,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7448–7458, Dec. 2020, doi: 10.1109/TII.2019.2958826.

[4] K. Hanbay, M. F. Talu, and Ö. F. Özgüven, “Fabric defect detection systems and methods—A systematic literature review,” *Optik*, vol. 127, no. 24, pp. 11960–11973, Dec. 2016, doi: 10.1016/j.ijleo.2016.09.110.

[5] Turgut Özseven, *“Surface Defect Detection and Quantification with Image Processing Methods”*, March 2019.

[6] J. Cao, G. Yang, and X. Yang, “A Pixel-Level Segmentation Convolutional Neural Network Based on Deep Feature Fusion for Surface Defect Detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021, doi: 10.1109/TIM.2020.3033726.

[7] D. Racki, D. Tomazevic, and D. Skocaj. A compact convolutional neural network for textured surface anomaly detection. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1331{1339, March 2018.

[8] Manpreet Singh Minhas, *“Anomaly Detection in Textured Surfaces”*, Canada, 2019.

[9] Hüseyin ÜZEN, Muammer TÜRKOĞLU, and Davut HANBAY, *“Result Weighting-Based Resnet Feature Pyramid Network Architecture for Surface Defect Detection”*, GU J Sci, Part C, 9(4):760-772 (2021).

[10] Benjamin Staara, Michael Lütjena and Michael Freitag, *“Anomaly detection with convolutional neural networks for industrial surface inspection”*, Procedia CIRP 79 (2019) 484–489.

[11] Oliver Rippel, Maximilian Müller dan Dorit Merhof, “GAN-based Defect Synthesis for Anomaly Detection in Fabrics”, *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020.

[12] Domen Tabernik, Samo Šela, Jure Skvarc dan Danijel Skocaj, “Segmentation-based deep-learning approach for surface-defect detection”, *Journal of Intelligent Manufacturing* (2020) 31:759–776.

[13] Umberto Michelucci, *“An Introduction to Autoencoders”*, Subjects: Machine Learning (cs.LG); Artificial Intelligence (cs.AI) , January 2022.

[14] Kaggle, “datasetpage,” 2021. [Online]. Avaliable: https://www.kaggle.com . [Accessed: 24 – Nov – 2022].

[15] M. J. Brusco and J. D. Cradit, “Graph Coloring, Minimum-diameter Partitioning, and the Analysis of Confusion Matrices,” *J. Math. Psychol*., vol. 48, no. 5, pp. 301–309, Oct. 2004.

[16] Xinyang Deng, Qi Liu, Yong Deng, Sankaran Mahadevan, *“An improved method to construct basic probability assignment based on the confusion matrix for classification problem”*, Information Sciences, 340-341, China (2016), pp. 250-261.