Atlantis Highlights in Engineering (AHE), volume 2
International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)

Multi-class Classification of Ceramic Tile Surface Quality using Artificial Neural Network and Principal Component Analysis

1st Muhammad Hanif Ramadhan
School of Industrial and System Engineering
Telkom University
Bandung, Indonesia
muh.hanif.ram@gmail.com

2nd Haris Rachmat
School of Industrial and System Engineering
Telkom University
Bandung, Indonesia
harisrachmat@telkomuniversity.ac.id

3rd Denny Sukma Eka Atmaja
School of Industrial and System Engineering
Telkom University
Bandung, Indonesia
dennysukma@telkomuniversity.ac.id

4th Rasidi Ibrahim
Mechanical and Manufacturing Engineering Faculty
Universiti Tun Hussein Onn
Johor, Malaysia
rasidi@uthm.edu.my

***Abstract*— The visual inspection of ceramic tile surface is an important factor which may influence the perceived surface quality of the product. While manual labor offers an alternative in the task of visual inspection, human limitation related problem such as fatigue and safety may pose an undesirable inspection performance when applied in mass production industry. This study attempted to automate the process of ceramic quality inspection through computerized image classification. Specifically, a dimensionality reduction technique called Principal Component Analysis and classification technique Artificial Neural Network were incorporated in the study to classify five categories of surface quality: normal, crack, chip-off, scratch and dry spots. Given 400 principal components as the input layer and three hidden layers consisting 150 hidden units each, the model was trained under 19,696 training images by using Adam Optimization. By performing prediction on the test set consisting of 4,256 images, the trained model was able to achieve the classification accuracy of 90.13%.**

Keywords—Artificial Neural Network, Industrial Visual Inspection, Principal Component Analysis, Surface Quality

I. INTRODUCTION

In ceramic tile's manufacturing industry, surface quality is one of the important aspects in determining the perceived quality of the product [1], [2]. One process that may influence surface quality of tiles is the process of visual inspection which involves rejecting tiles with unwanted faults within the surface [3]. In practice, the task of visual inspection is often done manually by operators [3]–[6] due to human capability to distinguish between accepted and defected tile surface. Unfortunately, utilization of human operators for visual inspection may hold several disadvantages. In the case where tiles are mass produced, it is laborious to manually inspect each surface of the tile. To handle this, sampling methods using Military Standards 105E was found to be an alternative inspection method [7]. However, the downside of using sampling method is the risk of letting defected product to pass through the inspection process [8]. Another disadvantage of manual inspection is related to human factors of the operator. In inspection task which require monotonous and repeating activities, the occurrence of human error would likely to increase due to fatigue and repetition [9]. Furthermore, The working environment of tiles inspection can be found to be hazardous and unhealthy which may also contributes significantly to human error [9].

In order to overcome the issues arise from manual quality inspection, an attempt to automate the visual inspection has been utilized in several studies by using Artificial Intelligence (AI) and Machine Learning techniques [5], [6], [10], [11]. A fuzzy logic system [12] with Gray-Level Co-Occurrence Matrix (GLCM) feature extraction was utilized by Putri et al. [6] to detect ceramic tile surface defect. The study was able to correctly classify 12 out of 13 test images and thus giving an accuracy rate of 92.31%. Sharma and Kaur [13] employed several machine learning techniques i.e. K-Nearest Neighbor, Support Vector Machine, and Bayesian Classifier to detect defect of ceramic tiles. Using 24 samples of test set, the study was able to get 70.84% of accuracy for each model.

Among Machine Learning algorithms, one of the models which capable of drawing complex non-linear decision boundaries is Artificial Neural Networks (ANN) [14]. Due to this property, extensive studies on the utilization of ANN approach on visual inspection system can be found in several literatures and researches [5], [11], [15], [16]. An example of such application can be found in [16] where a classification system using ANN was employed to automatically classify weld defects using 49 for testing examples with accuracy of 97.96%. Another related study [5] specifically in ceramic tiles inspection utilized ANN for surface defect detection by using GLCM feature extraction as preprocessing techniques. Given 32 samples as a training set and 13 samples for test set, an accuracy of 92.3% was obtained during the real time testing [5]. Furthermore, Mishra and Shukla [11] have employed Probabilistic Neural Network to ceramic tile inspection by trying to model the probability distribution of each class. With 50 tiles samples, the model was capable of reaching average accuracy of 98.20%.

While several studies found that ANN was a capable model for ceramic tile defects detection, further improvements can be done by evaluating the preprocessing stage carried out to the dataset [17]. Such improvement to the model can be achieved by finding a lower dimensional representation of the image while still keeping the important information consisted

within the dataset [18]. This can be achieved by one of the dimensionality reduction techniques called Principal Component Analysis (PCA) [17]. While reducing the dimensionality of the data, PCA can extract important features consisted in the dataset and compress the data size to reduce computational cost [17]. Several studies have employed PCA as a feature extraction technique for visual inspection [19], [20]. In one of the studies in leaf classification, PCA was found to outperformed the GLCM feature extraction with the accuracy of 98% and 78% respectively [21].

Given the problem of automatic visual inspection of ceramic tiles surface quality, this study aims to create a multi-class classification system which enables the classification of five different classes of ceramic tile surface: Normal, Crack, Dry Spots, Scratch, and Chip-off. This system was constructed by combining a dimensionality reduction technique i.e. PCA and non-linear classification model i.e. ANN.

II. Designing Ceramic Tile Surface Quality Classifier System via PCA and ANN

In order to construct a machine learning system which capable of surface defect classification and detection, this study propose a scheme which consist of three main stages: data pre-processing, training and hold out cross validation, and performance evaluation. Note that these stages were conducted in sequential manner as an overall process.

A. Data Pre-processing Stage

Using camera resolution of 1920x1080, each raw image of ceramic tile sample was captured at grayscale which consist of 1920x1080 different pixel values. For a given grayscale, image cropping with size of 50x50 was conducted across the tile image. The purpose of having 50x50 images patches as dataset was to enable the task object detection of the defected surface within the local region via sliding window algorithm [22]. Hence by applying this procedure throughout all samples, training set, validation set, and test set with the size of 2462, 521, and 532 respectively were acquired during the acquisition of images as shown in Table I.

TABLE I. Image patches acquired during the acquisition process

No	Class	Training set size	validation set size	Test set size
1	Normal	1,615	345	345
2	Chip off	144	33	41
3	Crack	269	26	26
4	Dry spots	366	111	96
5	Scratch	68	16	24
Total		2,462	531	532

In order make the number of dataset even larger, data augmentation was carry out to prevent the problem overfitting of the model by adding more artificial training examples [23]. Given the square window images of 50x50 pixels, augmentation was carry out by taking the rotational combinations (90̊, 180̊, 270̊) and transpose of each rotations (including transpose of the original images) as illustrated in Fig.1. Given the previous size of the original image, this process was able to increase the size of the dataset by the factor of 8 and therefore the training set, validation set, and test set size of 19,696, 4,248, and 4,256 respectively were obtained.

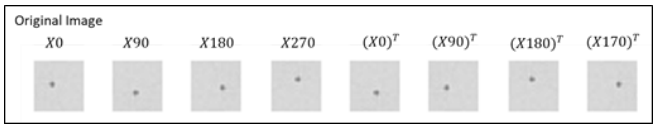

Fig. 1. Data Augmentation on a dryspot image

The next step in the preprocessing stage is to represent these image patches in the form of matrices. Given image patches of 50x50 pixels, the i[th] training case is represented by an vector of the unrolled version of the image $x^{(i)} \in \mathbb{R}^n$, where n is the number of pixels in the image i.e. 2,500. Each of $x^{(i)}$ is paired with label $y^{(i)} \in \{1,2,3,4,5\}$ which represent the surface quality. Therefore by stacking the whole m data points in column, a matrix of training set $X \in \mathbb{R}^{nxm}$ and vector $y \in \mathbb{R}^m$ are constructed from m data points (19,696 data points).

B. Training Stage

In this stage, training set and validation set from the augmentation result were utilized to train the machine learning system. The first step in this stage involving representation learning which conducted by PCA to reduce the dimensionality of the image patches. In the second step, the compressed representations of both training set and validation set are given to the ANN for training and hold-out cross validation procedure.

1) Dimensionality Reduction via PCA

Given the training set X, the output of PCA is to transform the dataset of each example of $x \in \mathbb{R}^n$ into a compressed representation of $z \in \mathbb{R}^k$ where k is the number of principal component such that $k \leq n$. This transformation can be obtain with minimal reconstruction error using simple matrix multiplication as in (1) [18]. Where $U \in \mathbb{R}^{nxn}$ is a matrix consisting principal component vectors stacked in column which can be obtained from the left singular vector of singular value decomposition (SVD) of the covariate matrix of X [18]. Thus, U_r is obtained by taking every element in first until the k-th column of the matrix U. Note that by computing SVD of the covariate matrix of X, a diagonal matrix S consisting the singular values $\lambda_{i,i}$ can be obtained as shown in (2) [24].

$$z^{(i)} = U_r^T x^{(i)} \quad (1)$$

$$S = \text{diag}\{\lambda_{1,1}, \lambda_{2,2}, \lambda_{3,3} \dots \lambda_{n,n}\} \quad (2)$$

$$\text{CPV}(k) = \frac{\sum_{i=1}^{k} \lambda_{i,i}}{\sum_{i=1}^{n} \lambda_{i,i}} \; x \; 100\% \quad (3)$$

One of the properties of SVD is that matrix S is constructed in such a way that $\lambda_{i,i}$ is placed from the greatest variance to the smallest. Consequently, the value of λ is also decreasing for each column towards the last column. Hence, in order to evaluate how much information is retained for a given number of principal component k, Cumulative Percent Variance (CPV) (3) was utilized to expressed the amount of variance retained [24].

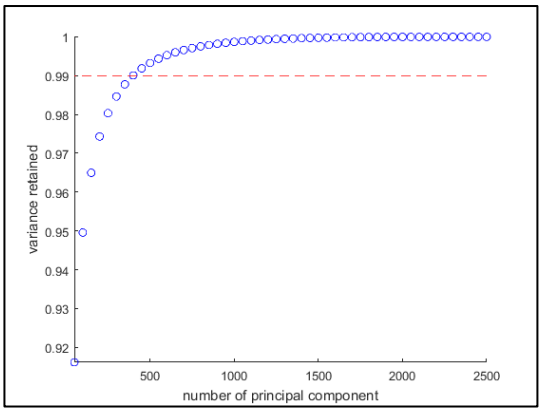

Fig. 2. Choosing the number of principal component k

The computation of CPV (3) was then performed under ranged values of k which was then plotted in Fig.2. Note that the result shown in Fig.2 revealed that the variance retained increase smoothly and flattens towards $k = 2500$. Hence given the plot, there is no clear boundary regarding what number of principal component k to keep. While increasing value of k lead to higher computational cost and noise, decreasing k to some extent would also throw away important information. One practical solution in determining the value of k is by choosing the smallest k in which able to keep the variance to some threshold value. In this study, a commonly used threshold of 99% variance retained was used to choose k [24]. As seen in Fig.4, the smallest number of principal component k that fulfil the criterion was 400 with 99% variance retained.

To evaluate how 400 principal components with 99% of variance retained may represent the original image patches of the ceramic quality, PCA is able to reconstruct the compressed representation into the approximation of the image. This approximation can be obtained by applying reconstruction function $g(z)$ as in [18]:

$$g(z^{(i)}) = U_r z^{(i)} \tag{4}$$

Given the reconstruction, visual comparison of the original and the approximation image was obtained as shown in Fig.3. Notice by retaining 99% of the variance, the reconstruction in Fig.3b was almost identical to the original image in Fig.3a. Hence, images in Fig.3.b can be interpreted as the representation that holds in the compressed representation $z \in \mathbb{R}^k$ where k is equal to 400.

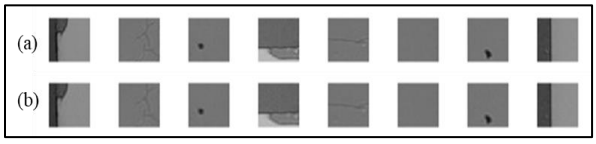

Fig. 3. PCA reconstruction: (a) Original Image (b) Reconstruction Image

2) ANN Training and Hold-out Cross-validation

In this study, feedforward and fully connected architecture of ANN [14] was utilized to perform the classification task as in Fig.4. Specifically, the architecture comprises three different kinds of layer: input layer, hidden layer, and output layer. Each layer l consists of artificial neurons. The number of neurons in the input layer and output layer were determined by the input dimensionality and the number of classes respectively. In contrast, the number of hidden layer and the number of neurons n_{l-1} in the hidden layer are hyperparameters which can be specified to affect the performance of the model.

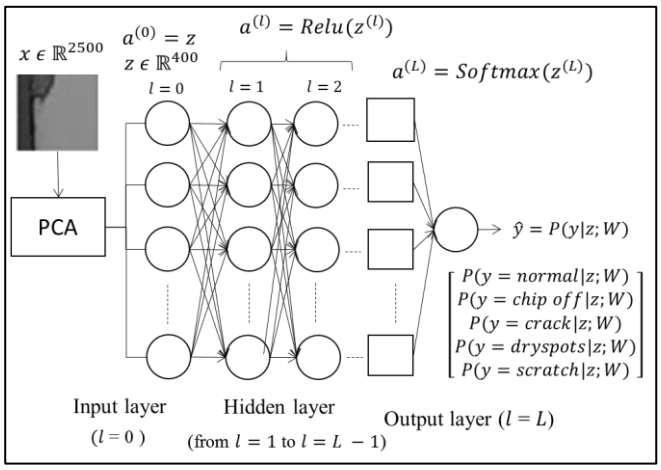

Fig. 4. Principal Component Analysis and Feedforward Neural Network architecture for ceramic curface quality classification

Given the lower dimensional representation of the image patches of tile's surfaces by PCA, training of ANN was conducted by using $z \in \mathbb{R}^k$ as the input layer of the network. The first step in generating prediction for given $z^{(i)}$ is to obtain the pre-activation function for a given layer by computing the weighted sum of over the number of neuron in the previous layer n_{l-1} as in (5)[14]:

$$\eta_j^{(l)} = \sum_{i=1}^{n_{l-1}} w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)} \tag{5}$$

Where $\eta_j^{(l)}$ is the j-th pre-activation function in layer l, $a_i^{(l-1)}$ is the activation function in $l-1$, $w_{ji}^{(l)}$ and $b_j^{(l)}$ is the weight connecting the i-th neuron in layer $l-1$ to the j-th neuron in layer l. Given $\eta^{(l)}$ as the pre-activation vector across all neurons in layer l, activation function $a_i^{(l)}$ is computed as the nonlinear transformation of $\eta^{(l)}$(6). Specifically, this study used the Rectified Linear Unit (ReLU) activation function as in (7) which was found to outperform log-sigmoid and tan-hyperbolic activations [14], [25]. To make the notation simpler, $a^{(0)}$ notation is assigned to the input layer of the network i.e. $a^{(0)}$ is equal to compressed representation z.

$$a^{(l)} = ReLU(\eta^{(l)}) \tag{6}$$

$$ReLU(\eta) = \max(0, \eta) \tag{7}$$

Finally in the output layer, $a^{(L)}$ or the prediction $\hat{y}$ is computed by an activation function called Softmax (8) which squash the pre-activation in the output layer $\eta^{(L)}$ into probability distribution of surface quality y given the compressed training case z i.e. $P(y|z;W)$ as in(8)[18] :

$$softmax(\eta^{(L)}) = \frac{\exp(\eta^{(L)})}{\sum_j^{n_L} \exp(\eta_j^{(L)})} \tag{8}$$

Thus, to make prediction given some input, forward propagation is applied by performing (5) then (6) for $l = 1$ to $l = L-1$ and (8) for $l = L$. Once Softmax compute the probability distribution of classes y given z, final classification was done by taking the class that has the highest probability $P(y|z;W,b)$.

Given the prediction $\hat{y} = P(y|z; W, b)$, the objective of the neural network can therefore be formulated as an unconstrained optimization which minimize some cost function $E(W, b)$ with respect of all weights W and biases b. In the case of classification by using softmax (8), cost function $E(W, b)$ can be defined as the cross-entropy loss which expressed as in (9) [18]. Where $\hat{y}_c$ is the probability of the true class target y_c given z. Equation (9) is then modified further by adding a weight decay regularization term to prevent overfitting, hence the regularized cost function $E_r(W, b)$ is given by (10) [26]. Where λ is the regularization parameter which influence the complexity of the decision boundary of the prediction $\hat{y}$. In order to obtain an appropriate parameters of W and b which minimize the cost function $E_r(W, b)$, some training procedure need to be conducted. One such method is by iterative learning that takes the gradients of the cost function $\partial E\left(w^{(l)}, b^{(l)}\right)/\partial w^{(l)}$ and $\partial E\left(w^{(l)}, b^{(l)}\right)/\partial b^{(l)}$ and use that information to update $w^{(l)}$ and $b^{(l)}$ for each layer. Specifically, in order to obtain $\partial E\left(w^{(l)}, b^{(l)}\right)/\partial w^{(l)}$ and $\partial E\left(w^{(l)}, b^{(l)}\right)/\partial b^{(l)}$ backpropagation was implemented on every iteration of training [27]. These gradients were then given as the information to calculate the first moment and the second moment estimates of Adam Optimization to update all $w^{(l)}$'s and $b^{(l)}$'s [28]. Hence, an iteration of training using Adam Optimization was carry out by performing forward propagation, backpropagation, and weights update according to the moment estimates.

$$E(W, b) = -\frac{1}{m}\sum_{i=1}^{m} \log\left(\hat{y}_c^{(i)}\right) \tag{9}$$

$$E_r(W, b) = E(W, b) + \frac{1}{m}\lambda \sum_{l=1}^{L}\sum_{j=1}^{n_l}\sum_{i=1}^{n_{l-1}} (w_{ji}^{[l]})^2 \tag{10}$$

The result of the training is shown in Fig.5 where the model was trained on three hidden layers ANN with 150 hidden neurons in each layer. Due to the considerably large training set of 19,696 images, the training was carry out by randomly dividing the training set into mini-batches of 1,024 images to reduce computational cost [29]. Hence, the cross entropy in Fig.5 appears to be fluctuating due to some mini-batches were slightly harder to predict that the other. Note that despite the fluctuations, the cross entropy was still revealing the sign of converging to the local optima.

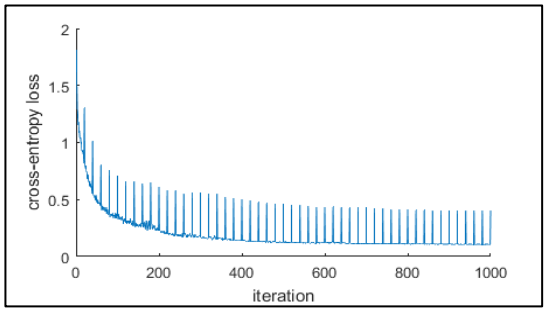

Fig. 5. Training result of Adam Optimization

In contrast with the internal parameters W and b, the regularization parameter λ is not learnable. One way to choose an appropriate parameter of λ was by performing a grid search over a hold-out validation set [18]. This can be performed by training the model several times over ranged values of λ. Specifically, a vector of logarithmically spaced values of λ's was generated. For each value of λ, training was performed and then evaluated by measuring the cross entropy of the training set and the validation set. The result of this procedure can be seen in Fig.6 where the optimal λ was found at 0.75 at the minimal validation loss. Notice that the training result in Fig.5 was carried out by using the optimal λ and therefore corresponds to the minimal validation loss.

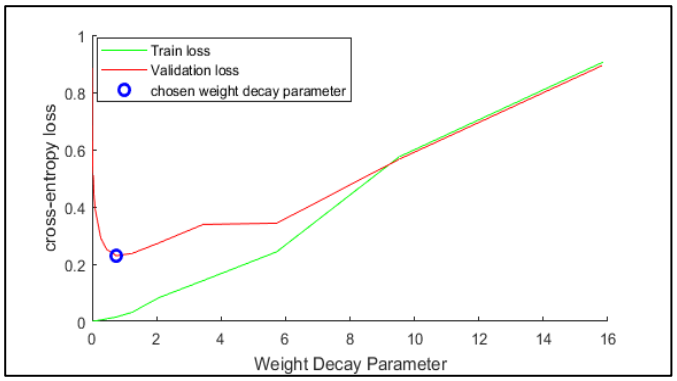

Fig. 6. Grid search over weight decay regularization parameter λ

C. *Performance Evaluation*

In the performance evaluation stage, predictions over the test set were employed to gain the accuracy of the model. The overall accuracy was obtained by calculating the percentage of the correctly classified labels over the whole test set. It was found that the model were able to reach the overall accuracy of 90.13% by correcly classified 3,836 test images out of 4,256 samples. Note that this evaluation metric is only assessing in terms of the number of correcly classified images over the whole samples in test set. Hence this metric does not evaluate the average performance in each class.

In order to assess the average performance of each class, this study used the second evaluation metric by calculating the macro-averaged F1- score [30]. The value of F1-score in each class can be obtained by calculating the harmonic mean of the precision P and recall R as in (11). By counting the number of true positives (tp), true negatives (tn), false positives (fp), and false negatives (tn) in the prediction, precision and recall can be obtained as in (12) and (13). Thus by averaging the value of F1-score of the five classes, the model was able to obtained the macro-averaged score of 0.78 out of 1.

$$F1 - score = \frac{2PR}{P + R} \tag{11}$$

$$P = \frac{tp}{tp + fp} \tag{12}$$

$$R = \frac{tp}{tp + fn} \tag{13}$$

III. CONCLUSIONS

An attempt to automate visual inspection of ceramic tile surface quality was conducted by applying an image multi-class classification technique. In this study, the combination of PCA and ANN were proposed in order to classify 5 distinct classes of ceramic tile surface quality: normal, chip-off, crack, dry spots and scratch. The creation of such model was

carried out in three main stages: data pre-processing, training and hold-out cross validation, and performance evaluation.

In the pre-processing steps, data augmentation was employed to the dataset consisting grayscale image patches of ceramic surface. By performing simple rotation and transpose operations, training set of 19,696 images was obtained in the augmentation process. By using larger dataset, the model was able to generalize better due to the regularization effect of the augmentation.

The training was started by the representation learning of PCA which able to reduce the amount of dimensionality from 2,500 to 400 with 99% variance retained. Due to the relatively large size of dataset, the training in the batch manner was computationally expensive due to large matrix size needed to be processed in each update. Hence this study also utilized Adam Optimization under mini-batch update. The training was performed under a three hidden layer neural network with 150 hidden units in each layer. With the weight decay regularization of 0.75 obtained from the hold-out cross validation, the result shows that the optimization was able to find the point near the local optima.

Given the parameters resulted from the training, test images were than employed to the model to evaluate the performance. The proposed model was able to achieve an overall accuracy of 90.13% on 4,256 test images were obtained with macro-averaged F1-score of 0.78. Thus given the results of the model, this study is able to provide an alternative method for visual inspection of ceramic tiles surface quality.

REFERENCES

[1] L. J. Callarisa Fiol, M. A. Moliner Tena, J. Sánchez García, and ., "Multidimensional perspective of perceived value in industrial clusters," *J. Bus. Ind. Mark.*, vol. 26, no. 2, pp. 132–145, 2011.

[2] S. Singh and M. Kaur, "Machine Vision System for Automated Visual Inspection of Tile's Surface Quality," *IOSR J. Eng.*, vol. 2, no. 3, pp. 429–432, 2012.

[3] H. Elbehiery, A. Hefnawy, and M. Elewa, "Surface Defects Detection for Ceramic Tiles Using Image Processing and Morphological Techniques," *Proc. World Acad. Sci. Eng. Technol. Vol 5*, vol. 5, no. 5, pp. 158–162, 2007.

[4] D. S. E. Atmaja and M. K. Herliansyah, "Optimasi Proses Pengukuran Dimensi Dan Defect Ubin Keramik Menggunakan Pengolahan Citra Digital Dan Full Factorial Design," *J. Teknosains*, vol. 4, no. 2, pp. 179–191, 2015.

[5] P. A. Lestari, H. Rachmat, and D. S. E. Atmaja, "Design of Automation System for Ceramic Surface Quality Control Using Artificial Neural Network at Balai Besar Keramik," *Univ. Telkom, S1 Tek. Ind.*, vol. 4, no. 2, pp. 2746–2753, 2017.

[6] A. P. Putri, H. Rachmat, and D. S. E. Atmaja, "Design of Automation System for Ceramic Surface Quality Control Using Fuzzy Logic Method At Balai Besar Keramik (Bbk)," *Univ. Telkom, S1 Tek. Ind.*, vol. 4, no. 2, pp. 2515–2520, 2017.

[7] A. D. Septiani and D. Anne, "Evaluasi Pelaksanaan Sistem Pengendalian Kualitas pada PT. X," *J. Titra*, vol. 1, no. 1, pp. 33–40, 2013.

[8] D. Montgomery, *Introduction to statistical quality control*. 2009.

[9] J. A. Yeow, P. K. Ng, K. S. Tan, T. S. Chin, and W. Y. Lim, "Effects of Stress, Repetition, Fatigue and Work Environment on Human Error in Manufacturing Industries," *J. Appl. Sci.*, vol. 14, no. 24, pp. 3464–3471, 2014.

[10] A. O. M. Luiz, L. C. P. Flavio, and E. M. A. Paulo, "Automatic detection of surface defects on rolled steel using Computer Vision and Artificial Neural Networks," *IECON 2010 - 36th Annu. Conf. IEEE Ind. Electron. Soc.*, no. c, pp. 1081–1086, 2010.

[11] R. Mishra and D. Shukla, "An Automated Ceramic Tiles Defect Detection and Classification System Based on Artificial Neural Network," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 4, no. 3, pp. 229–233, 2014.

[12] H. Rachmat, T. Mulyana, S. B. H. Hasan, and M. R. Bin Ibrahim, "Design selection of In-UVAT using MATLAB fuzzy logic toolbox," in *Advances in Intelligent Systems and Computing*, 2017.

[13] M. Sharma and G. Kaur, "Integrated Approach for Defect Detection in Ceramic Tiles," *Int. J. Comput. Technol.*, vol. 3, no. 2, pp. 259–262, 2012.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] S. Bhuvaneswari and J. Sabarathinam, "Defect Analysis Using Artificial Neural Network," *Int. J. Intell. Syst. Appl.*, vol. 5, no. 5, pp. 33–38, 2013.

[16] T. Y. Lim, M. M. Ratnam, and M. A. Khalid, "Automatic classification of weld defects using simulated data and an MLP neural network," *Insight*, vol. 49, no. 3, pp. 154–159, 2007.

[17] H. Abdi and L. J. Williams, "Principal Component Analysis," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–470, 2010.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, vol. 521, no. 7553. 2016.

[19] J. Mirapeix, P. B. García-Allende, A. Cobo, O. M. Conde, and J. M. López-Higuera, "Real-time arc-welding defect detection and classification with principal component analysis and artificial neural networks," *NDT E Int.*, vol. 40, no. 4, pp. 315–323, 2007.

[20] N. Zuber and R. Bacjric, "Application of artificial neural networks and principal component analysis on vibration signals for automated fault classification of roller element bearings," *Eksploat. i Niezawodn. - Maint. Reliab.*, vol. 18, no. 2, pp. 299–306, 2016.

[21] A. Ehsanirad and S. K. Y. H, "Leaf recognition for plant classification using GLCM and PCA methods," *Comput. Sci. Technol.*, vol. 3, no. 1, pp. 31–36, 2010.

[22] G. Papandreou, I. Kokkinos, P. A. Savalle, and Dummy, "Modeling local and global deformations in Deep Learning: Epitomic convolution, Multiple Instance Learning, and sliding window detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07–12–June, pp. 390–399, 2015.

[23] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," *Unpublished*, 2017.

[24] S. Valle, W. Li, and S. J. Qin, "Selection of the Number of Principal Components : The Variance of," *Society*, pp. 4389–4401, 1999.

[25] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *AISTATS '11 Proc. 14th Int. Conf. Artif. Intell. Stat.*, vol. 15, pp. 315–323, 2011.

[26] A. Krogh and J. A. Hertz, "A Simple Weight Decay Can Improve Generalization," *Adv. Neural Inf. Process. Syst.*, vol. 4, pp. 950–957, 1992.

[27] T. M. Mitchell, *Machine Learning*, vol. 1, no. 3. 1997.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Conf. Pap. ICLR 2015*, pp. 1–15, 2014.

[29] M. Morishita, Y. Oda, G. Neubig, K. Yoshino, K. Sudoh, and S. Nakamura, "An Empirical Study of Mini-Batch Creation Strategies for Neural Machine Translation," pp. 61–68, 2017.

[30] A. Özgür, L. Özgür, and T. Güngör, "Text Categorization with Class-Based and Corpus-Based Keyword Selection," *Proc. 20th Int. Conf. Comput. Inf. Sci.*, vol. 3733, pp. 606–615, 2005.