ICIC Express Letters
Volume **15**, Number **2**, February **2021**

ICIC International ©2021 ISSN 1881-803X
pp. 157–164

FACE IDENTIFICATION USING MULTI-LAYER PERCEPTRON AND CONVOLUTIONAL NEURAL NETWORK

Kartika Fithriasari and Ulfa Siti Nuraini

Department of Statistics
Institut Teknologi Sepuluh Nopember
Sukolilo, Surabaya Regency, Jawa Timur 60111, Indonesia
kartika_f@statistika.its.ac.id; nuraini.19062@mhs.its.ac.id

Received June 2020; accepted September 2020

Abstract. *Algorithm that is often used for image processing is machine learning. One of machine learning is Neural Network, which has additional layer called Multi-Layer Perceptron (MLP). Besides, other technique is Convolutional Neural Network (CNN). In this paper, we compare MLP and CNN methods carried out in image processing for face identification case study. We used preprocessing to change RGB to grayscale and normalize it. We proposed to use image augmentation to get more data without taking some images again. Lack of images can make overfitting so image augmentation may reduce it. For training, we used Adam optimizer. It is more efficient to train images that have complicated patterns. The conclusion is that CNN method has better results, due to its higher accuracy, precision, sensitivity, and Fscore than MLP method.*

Keywords: Convolutional neural network, Face, Image, Multi-layer perceptron

1. **Introduction.** Developing technology, data are not always in numbers structural form that has been constructed at the beginning. However, data can also be images, sounds, and others. Face identification is one of image processing. There are many methods for it, but most of them are included in machine learning. With simple language, machine learning is as computer algorithms to study data, recognize patterns, and make models based on historical data. Neural Network (NN) is one part of machine learning. It is a form of artificial intelligence that can learn from data and capture nonlinear pattern [1]. Type of NN model which consists of many layers is called Multi-Layer Perceptron (MLP) which connects fully between neurons and has a powerful classification ability. One research conducted by Singh and Sachan [2] in a case study identified the handwriting of Gurmukhi characters (Indian script) using MLP. Other cases are done by Zahrati et al. [1] and Fithriasari et al. [3]. Another technique that has significant results in image recognition is the Convolutional Neural Network (CNN) [4]. That is because CNN tries to imitate the image recognition system in the human visual cortex so that it can process image information. CNN's ability is claimed as the best model for solving object detection and object recognition problems [5]. Some studies are done by Tivive and Bouzerdoum [6], Fu et al. [7] and Zhao et al. [8].

A comparison of the MLP and CNN methods was done by Medina et al. [9] to detect algae classification in the pipeline, which gets the best method is obtained by CNN which gives a higher accuracy than MLP. In this paper, we applied comparison MLP and CNN with another case study, and it is about biometric system using face. We purposed to use data augmentation and Adam optimizer. Data augmentation in deep learning is done by Mikolajczyk and Grochowski [10] for image classification. We used data augmentation to get more data without taking more photo, so it can be simple for getting new data. Lack of data can make overfitting, so data augmentation will reduce it. Adam optimizer used

DOI: 10.24507/icicel.15.02.157

for training is combination of RMSprop that works with rescaled gradient and SGD that works with momentum, so we get not only less cost but also less iterations than other optimizers like AdaGrad, SGD, and RMSprop [11].

Academic achievement of this paper is that we can provide other studies to compare MLP and CNN methods so that we can contribute to the world of research in the development of face identification. Another achievement is to get best model that we can use in GUI to record attendance with a biometric system. The goal in this paper is to get the best method between MLP and CNN in face identification. After getting the best method, Graphic User Interface (GUI) will be created that can identify faces.

2. Reference.

2.1. Digital image and image augmentation.

Digital image is digital data on a computer that represents an image. Digital images can be described as $f(x, y)$ functions where x and y are coordinates on a flat plane that represents a collection of pixels in two dimensions [12]. An RGB color image is the original image that is captured by camera. In RGB type images, each pixel has 3 color components, namely red – R, green – G, and blue – B. Each color component has a range of values between 0 and 255. Grayscale type is images that only have one value per pixel as gray degrees. Gray degree has a value between 0 (black) to 255 (white) [12]. The process of changing RGB into grayscale can be done by weighting on each color component R, G, and B [13] in Equation (1).

$$gray = 0.299R + 0.587G + 0.114B \tag{1}$$

Image processing that is used for adding training data is called image augmentation. Image augmentation can reduce overfitting [14]. Traditional transformation (like rotate, flip, and zoom) is augmentation because of combination. Image enhancement (like brightness, sharpening, contrast, and pseudo-coloring) is augmentation because of manipulating parameters image that accentuate special characteristics. Because of image augmentation we can make more images based on one image without we took another image. It makes more efficient to analyze data.

2.2. Multi-layer perceptron.

Multi-Layer Perceptron (MLP) is a method on the neural network where the model includes hidden layer and weight, and inside hidden layer using activation function. Activation function is a function that describes the relationship between inputs to issue output values that may be linear or non-linear [15]. Basically, MLP consists of 3 layers, namely the input layer, hidden layer, and the output layer. In MLP, it can vary the number of hidden layers and the number of neurons within hidden layer [16].

Each input that is connected to each neuron in the hidden layer and the output layer, has a bias and weight. Equation (2) is an equation in hidden layer neurons without an activation function, and then the results of Equation (2) are included in the ReLU (Rectified Linear Unit) activation function with Equation (3). Then a calculation is made to connect to the output layer with Equation (4) and to get the predicted value, and the result of Equation (4) is given the SoftMax activation function with Equation (5). Finally, calculate to get the classification results. The target or class obtained is a decision of predicted value that does not always produce a rounded value $(1, 2, \ldots, k)$ but can be in the form of an opportunity value by determining the class through the threshold of the researcher [15].

$$p_j = a_j + \sum_{j=1}^{n_h} w_{i,j} X_i \tag{2}$$

$$q_j = f_h(p_j) = \max(0, p_j) = \begin{cases} p_j, & \text{if } p_j \geq 0 \\ 0, & \text{if } p_j < 0 \end{cases} \tag{3}$$

$$r_k = b_k + \sum_{k=1}^{K} v_{j,k} q_j \tag{4}$$

$$\hat{y}_k = f_o(r_k) = \frac{\exp(r_k)}{\sum_{k=1}^{K} \exp(r_k)} \tag{5}$$

2.3. **Loss and Adam optimization parameter.** Loss is a function used as a criterion that must be minimized. Minimum loss can be used as a solution to get best model while comparing models [17]. One of losses is cross-entropy in Equation (6), and it is calculated from the error between two probabilities which are produced by SoftMax [18].

$$Loss = -\sum_{k=1}^{K} y_k \ln(\hat{y}_k) \tag{6}$$

Parameter optimization is used to minimize the value of loss, so loss is the key in optimizing the parameters of bias and weight. The parameter optimization used in this research is Adam's parameter optimization. Adam (adaptive moment estimation) is an adaptive training optimization algorithm designed specifically for training in deep learning methods. Adam uses a gradient, then estimates the first and second moments, and corrects with bias correction [11].

2.4. **Convolutional Neural Network (CNN).** CNN follows the basic assumption of Neural Network (NN), but in CNN not all neurons have full connectivity, except at the fully connected layer. In other words, CNN transforms the original layer by layer image from the pixel value of the image into a class scoring value for classification. The basic layers used in CNN are convolutional, pooling, and fully connected layer [19]. Convolution in image processing is one method for obtaining feature extraction from an image. The convolution process is to multiply an image with a convolution kernel or filter expressed in the form of a matrix. ReLU in Equation (3) is activation function that is used between convolution layer and next layer. Pooling layer is used to ensure that the image only focuses on certain patterns that characterize the image, so it can reduce dimensions. Fully connected layer is used to make into 1 vector and classified linearly like multi-layer perceptron [19]. Activation function used in output layer is SoftMax in Equation (5).

3. **Methodology.** In this paper, we used primary data, taken from webcam images with dimensions of 640 × 480 pixels. Data are photos of 4 people that each person is taken 10 photos from different sides. For each photo, we resize to 160 × 120 and change image from RGB to grayscale, and then we normalize grayscale value to 0 and 1. Using more image inputs, it will provide better learning for making models. Therefore, we use image augmentation or data multiplication for each image listed in Figure 1.

Figure 1. Image augmentation

In this paper, we used 10-fold cross validation with a ratio between the number of training and testing is 288:32. After that, we are classifying using MLP method and CNN method. For training phase in each method and each fold, activation function in output layer is SoftMax with Adam optimizer. The iteration process will stop if the value of loss is less than the specified minimum loss (set of 0.05) or more than the maximum number of iterations (epoch) (set of 300). Then, we compute performance classification in every fold. Best method is obtained from the average performance classification for ten folds. It will be used in Graphic User Interface for classifying new image. Last, we take conclusions and suggestions.

4. **Analysis and Discussion.**

4.1. **Characteristics data.** Preprocessing data is used to change RGB image to grayscale, and then resize it to 160×120. The histogram of example image for each class after preprocessing is shown in Figure 2. It can be shown that histogram between one image and another is not much different. There are 2 patterns for each histogram, such as black to gray and gray to white, because of background. However, in this study, we keep using background for training.

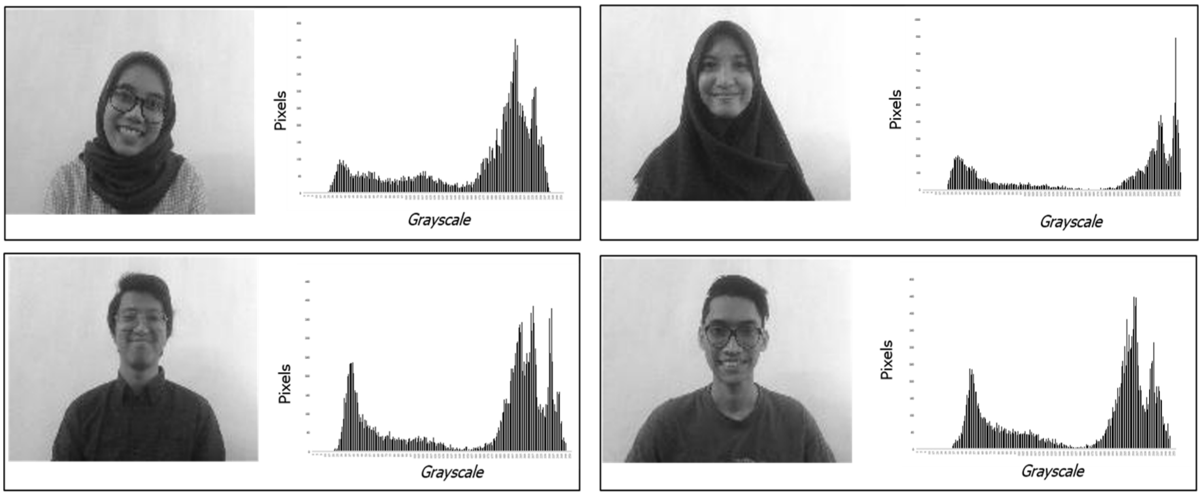


FIGURE 2. Histogram of one image for each class

4.2. **Multi-layer perceptron.** The results of preprocessing data obtained 19,200 ($160 \times 120 = 19{,}200$) input variables. The experiments were carried out on 1 hidden layer and 2 hidden layers with each hidden layer having 60-150 units. The output variable is 4, equal to the number of people who took part in. Classification performance results obtained in ten folds with one and two hidden layers are shown in Table 1. The best classification performance on 1 hidden layer is with 90 nodes. Unlike 2 hidden layers, the best classification performance results are found with 120 nodes. From this case, it was

TABLE 1. Performance classification MLP

Nodes	1 hidden layer				2 hidden layers			
	Acc	Pre	Sens	Fsc	Acc	Pre	Sens	Fsc
60	0.9313	0.9371	0.9313	0.9342	0.9281	0.9338	0.9281	0.9309
70	0.9313	0.9384	0.9313	0.9348	0.9250	0.9309	0.9250	0.9279
80	0.9313	0.9371	0.9313	0.9342	0.9250	0.9309	0.9250	0.9279
90	**0.9406**	**0.9463**	**0.9406**	**0.9434**	0.9281	0.9341	0.9281	0.9311
100	0.9281	0.9352	0.9281	0.9317	0.9313	0.9375	0.9313	0.9343
110	0.9344	0.9406	0.9344	0.9375	0.9250	0.9309	0.9250	0.9279
120	0.9219	0.9272	0.9219	0.9245	**0.9344**	**0.9402**	**0.9344**	**0.9373**
130	0.9313	0.9380	0.9313	0.9346	0.9250	0.9317	0.9250	0.9283
140	0.9344	0.9401	0.9344	0.9372	0.9281	0.9338	0.9281	0.9309
150	0.9250	0.9307	0.9250	0.9278	0.9250	0.9306	0.9250	0.9278

found that more nodes do not always increase the classification performance. Also, more hidden layers do not always increase the classification performance.

Further, we look at each fold to get the classification performance on every fold. Figure 3 shows the result of the classification performance of each fold on 1 hidden layer for 90 nodes.

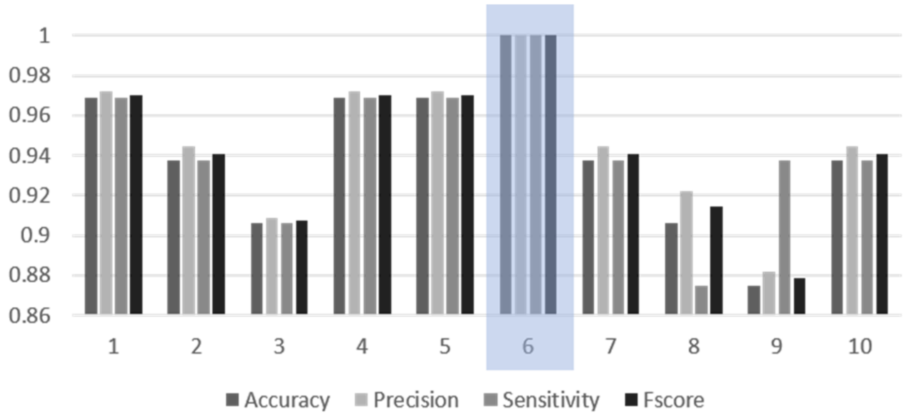


FIGURE 3. Performance classification of MLP for each fold

Figure 3 shows the best classification performance value is the 6th fold which is the largest value. Equation that we get in the 6th fold is based on Subsection 2.2, in Equations (2) until (5). This is the equation between the input layers for each node in the hidden layer according to Equation (2).

$$p_1 = -0.02635x_1 + 0.01238x_2 - 0.02522x_3 + \cdots - 0.03239x_{19200},$$
$$\ldots$$
$$p_{90} = -0.02887x_1 - 0.04364x_2 - 0.00854x_3 + \cdots - 0.02806x_{19200}.$$

Furthermore, the equation for each node in the hidden layer with the ReLU activation function based on Equation (3) is as follows.

$$q_1 = f_h(p_1), q_2 = f_h(p_2), \ldots, q_{90} = f_h(p_{90}).$$

The equation in the output layer before the activation function is entered based on Equation (4) is as follows.

$$r_1 = -0.02427 - 0.02635q_1 - 0.03152q_2 + \cdots - 0.0173q_{90},$$
$$\ldots$$
$$r_4 = 0.03028 + 0.03057q_1 - 0.01641q_2 + \cdots - 0.03878q_{90}.$$

So, the equation for each node in the output layer with the SoftMax activation function based on Equation (5) is as follows.

$$\hat{y}_1 = f_o(r_1), \hat{y}_2 = f_o(r_2), \hat{y}_3 = f_o(r_3), \hat{y}_4 = f_o(r_4).$$

The result of the equation is between 0 and 1. To determine the resulting prediction class k, it is seen from the maximum value. The best model illustration is in Figure 4.

4.3. **Convolutional neural network.** In this subsection, we compare filter size of convolutional layer and pooling layer. The number of nodes used in fully connected layer is 10 nodes. The performance results presented in Table 2. This table shows average of 10 folds performance classification. The best classification performance is on model 9×9 convolutional layer and 8×8 pooling layer filter size. The multiplication of filter sizes in each layer does not indicate an increase in classification performance.

Figure 5(a) results from the classification performance of each fold using the CNN method. It shows that the best amount to 3 folds are 4, 5, and 9. To get the best model

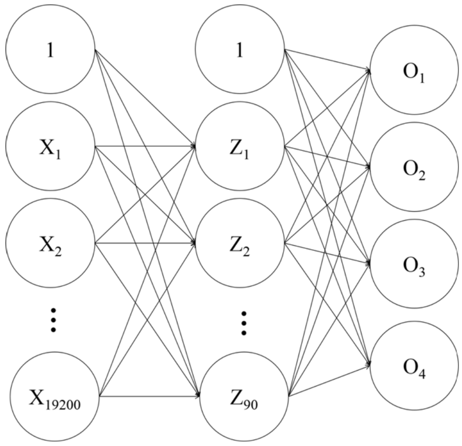
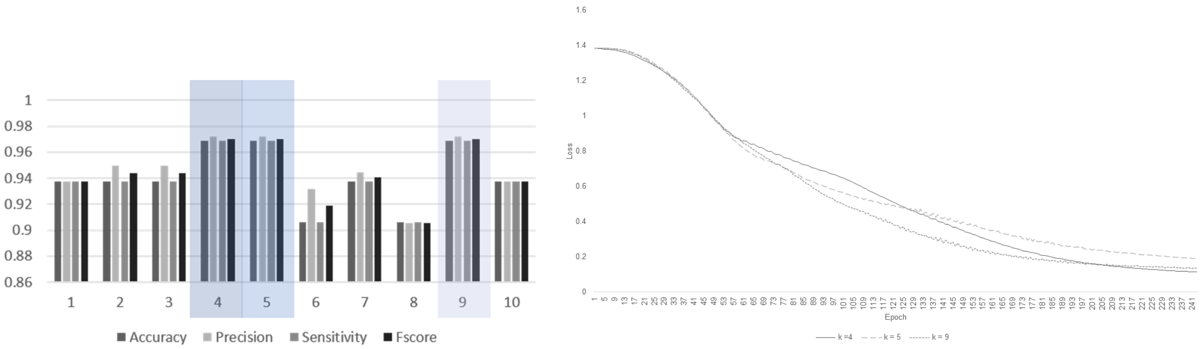


FIGURE 4. Illustration of MLP

TABLE 2. Performance classification of CNN

Conv	Pool	Accuracy	Precision	Sensitivity	Fscore
3×3	2×2	0.9250	0.9336	0.9250	0.9292
5×5	2×2	0.9344	0.9388	0.9344	0.9366
5×5	4×4	0.9313	0.9367	0.9313	0.9340
6×6	5×5	0.8844	0.8619	0.8844	0.8717
7×7	2×2	0.9219	0.9289	0.9219	0.9254
9×9	2×2	0.9063	0.9141	0.9063	0.9102
9×9	4×4	0.9281	0.9333	0.9281	0.9307
9×9	**8×8**	**0.9406**	**0.9473**	**0.9406**	**0.9439**
11×11	2×2	0.8969	0.9059	0.8969	0.9014
11×11	5×5	0.9375	0.9436	0.9375	0.9405
11×11	10×10	0.9375	0.9426	0.9375	0.9400



(a) (b)

FIGURE 5. (a) Performance classification of MLP for each fold; (b) graph of loss

then proceed to see the value of loss in the fold to determine the fold that has a minimum loss value. Figure 5(b) is a graph of loss for the 4th fold, 5th fold, and 9th fold. From the graph, the fastest loss reduction is for the 9th fold, but the loss value for the 4th fold has a value that is smaller than the 9th fold. In addition, the number of epochs used in the training process is more on the 4th fold so that the losses obtained tend to be smaller. So the best fold for the convolutional neural network method is with a 9×9 convolutional filter and an 8×8 pooling filter is the 4th fold. Figure 6 shows an illustration of the CNN model with the best classification performance.

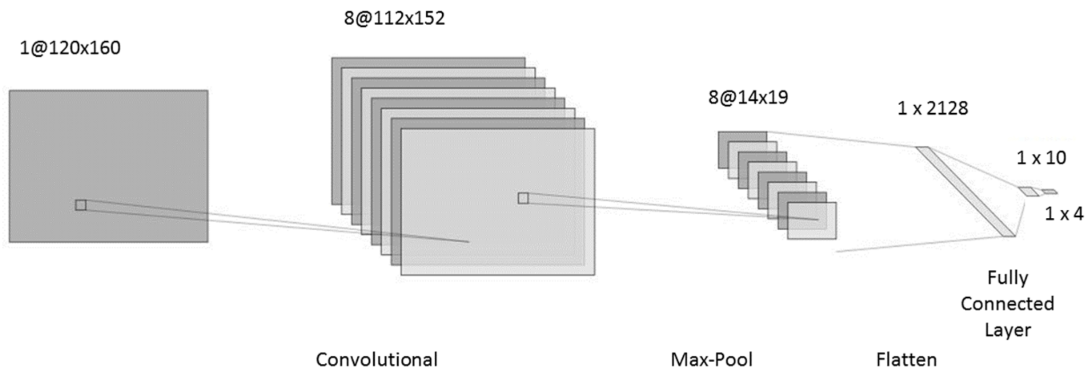


Figure 6. Illustration of CNN

4.4. **Comparison between MLP and CNN.** Comparison between MLP and CNN can be done with loops several items. In this section, the models we want to compare are MLP with 1 hidden layer 90 neurons in the 6th fold and CNN with the 8th model in the 4th fold. The results of average of 10 loops performance classification are shown in Table 3. Method that has high average performance is CNN. All performances are getting up 90%, so this method is very good for classification. CNN is simpler than MLP, and it is obtained from number of parameters. More parameters can make hard for training and make complicated models. So, we can take conclusion that CNN has better result than MLP.

Table 3. Comparison of CNN and MLP

	MLP	CNN
Accuracy	0.9031	0.9750
Precision	0.9243	0.9778
Sensitivity	0.9031	0.9750
Fscore	0.9136	0.9764
Number of parameters	1,728,454	9,172

5. **Conclusions.** Based on the analysis and discussion, the conclusion is that a method that has better performance is convolutional neural network method with a 9×9 convolutional filter and an 8×8 pooling filter. We get performance of CNN is better than MLP; one of the reasons is that CNN is simpler than MLP. It can be seen from the parameter used for training. For further research, facial cropping can be done so that it can be centered on the face and reduce noise. Using different number of nodes and filter possible can accept more useful result.

REFERENCES

[1] Z. Zahrati, K. Fithriasari and Irhamah, Multi-output neural network for the temperature forecasting in Semarang, *AIP Conference Proceedings,* vol.1746, 020040, 2016.

[2] G. Singh and M. Sachan, Multilayer perceptron neural network technique for offline handwritten Gurmukhi character recognition, *IEEE International Conference*, Coimbatore, India, 2014.

[3] K. Fithriasari, N. Iriawan, B. S. S. Ulama and Sutikno, On the multivariate time series rainfall modeling using time delay neural network, *International Journal of Applied Mathematics and Statistics*, vol.44, no.14, pp.193-201, 2013.

[4] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, Tokyo, Japan, 1980.

[5] A. Karpathy, *CS231n: Convolutional Neural Network for Visual Recognition*, http://cs231m.github.io/, 2018.

[6] F. H. C. Tivive and A. Bouzerdoum, A gender recognition system using shunting inhibitory convolutional neural networks, *International Joint Conference on Neural Networks*, New York, USA, 2006.

[7] J. Fu, Y. Huang, J. Xu and H. Wu, Optimization of distributed convolutional neural network for image labeling on asynchronous GPU model, *International Journal of Innovative Computing, Information and Control*, vol.15, no.3, pp.1145-1156, 2019.

[8] J.-Y. Zhao, J. Gong, S.-T. Ma and Z.-M. Lu, Curvature gray feature decomposition based finger vein recognition with an improved convolutional neural network, *International Journal of Innovative Computing, Information and Control*, vol.16, no.1, pp.77-90, 2020.

[9] E. Medina, M. R. Petraglia, J. G. R. C. Gomes and A. Petraglia, Comparison of CNN and MLP classifiers for algae detection in underwater pipelines, *2017 the 7th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Brazil, 2017.

[10] A. Mikolajczyk and M. Grochowski, Data augmentation for improving deep learning in image classification problem, *2018 International Interdisciplinary PhD Workshop (IIPhDW) Conference*, Swinoujscie, Poland, 2018.

[11] D. P. Kingma and J. L. Ba, Adam: A method for stochastic optimization, *International Conference on Learning Representations 2015*, San Diego, USA, 2015.

[12] R. E. Woods and R. E. Gonzales, *Digital Image Processing*, 3rd Edition, Pearson Education, New Jersey, 2008.

[13] Faculty of Chemistry, Brno University of Technology, *Image Science Fundamentals*, https://www2.fch.vut.cz/lectures/imagesci/, 2019.

[14] J. Wang and L. Perez, The effectiveness of data augmentation in image classification using deep learning, *Convolutional Neural Networks Vis. Recognit.*, 2017.

[15] C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, *Activation Functions: Comparison of Trends in Practice and Research for Deep Learning*, www.arxiv.com, 2018.

[16] W. S. Sarle, Neural networks and statistical models, *The 19th Annual SAS Users Group International Conference*, NC, USA, 1994.

[17] R. Russell and R. J. Marksll, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, The MIT Press, Cambridge, 1999.

[18] I. GoodFellow, Y. Bengio and A. Courville, *Deep Learning*, The MIT Press, Cambridge, 2016.

[19] A. Ferreira and G. Giraldi, Convolutional neural network approaches to granite tiles classification, *Expert System Application*, vol.84, pp.1-11, 2017.