

CLASSIFICATION OF TILES USING CONVOLUTIONAL NEURAL NETWORK

¹Susanti Aulia Ramadayanti, ²Adhi Prahara

^{1,2}Department of Informatics, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

e-mail: ¹*susanti1700018179@webmail.uad.ac.id, ²adhi.prahara@tif.uad.ac.id

*correspondence email

Abstract

Tiles are one of the building materials with various types that can make a residence more elegant, attractive, and colorful. However, not all people know about the types of tiles and their advantages. Therefore, a Convolutional Neural Networks (CNN) based method is proposed to make it easier for people to accurately recognize tiles based on their types and know their advantages. The purpose of this paper is to classify the types of tiles using CNN which is based on VGG16 model. The proposed method classifies tiles into 6 classes, namely granite, limestone, marble, motifs, mosaics, and terrazzo. This research uses 186 training data, 96 validation data and 60 test data with image resolution of 224x224. Based on the experiments, the training process produces 100% of training accuracy and 94% of validation accuracy. The testing process achieves 98.33% accuracy which can be concluded that the proposed CNN model able to classify the types of tiles well.

Keywords: Classification, Tiles, CNN, VGG16

INTRODUCTION

Tiles are craft objects made of non-metallic elements that have undergone a heating process in their manufacture. The process of manufacturing, for example ceramic tile consists of several steps namely clay preparation, molding, glaze preparation, drying, glazing and decorate the tile, kiln firing, and quality classification [1]. Selection of the tiles is in accordance with the needs of the buildings. The process of selecting tiles usually based on the shape, size, and color of the tiles. This study aims to determine the type of tiles based on the images provided by the user.

Current technological developments can be applied to solve the tile classification problem. Using computer vision approach, the first step of the procedure is to take pictures of tiles with a camera. The images will be classified using Convolutional Neural Networks (CNN) to recognize the type of tiles [2]. CNN was used because several recent studies on computer vision task that use CNN have obtained good results, namely research that classify vehicle types [3], batik patterns [4], face [5], leather types [6][7] and so on [8][9]. This research was conducted because there are many types of tiles so not all people know about the types of tiles and their advantages. There are several types of tiles such as ceramic, natural granite, marble, mosaic, porcelain, glass, cement, limestone, and others with different advantages [10]. However, in this study only 6 types of tiles were used, namely terrazzo, limestone, natural granite, motif, mosaic, and marble because these types of tiles are popular and widely used in general.

In this study, VGG16-based CNN model is implemented, namely, to classify the types of tiles. CNN method is very good at finding good features in the next layered image to form a non-linear hypothesis and can increase the complexity of a model [11]. Accuracy results in image classification using CNN are mostly better than the other traditional methods such as K-Nearest Neighbor, Support Vector Machine and others [12][13][14].

METHODS

Convolutional Neural Network (CNN)

Convolutional Neural Network is a branch of deep learning that carry out direct classification learning on two-dimensional models such as images [15]. CNN is very good when used to find patterns in an image then recognize the object in the image. CNN consists of two stages, namely feature learning and classification [16]. Feature learning is the stage where the features of input image are extracted to learn the value/pattern of the image [17]. At this feature learning stage, it is very dependent on the depth of the convolutional layers, the deeper the convolutional layer, the more extraction results are obtained so that the pattern is clearer [18]. This value will be converted into a vector and used in the classification stage. In the classification stage, some fully connected layers will be used to classify the features into several classes [19].

Fig. 1 shows the Convolutional Neural Network architecture that divided into two stages. The feature learning stage in general has three layers namely convolution layer, activation, and pooling layer. The sequence of these three layers does not always have to be the same, in the sense that the process can be modified as needed. However, in general, the feature learning process begins with a convolution process between the input matrix and the kernel at a certain size. The classification layers usually consist of several fully connected layers with the last layer serves as the predictor.

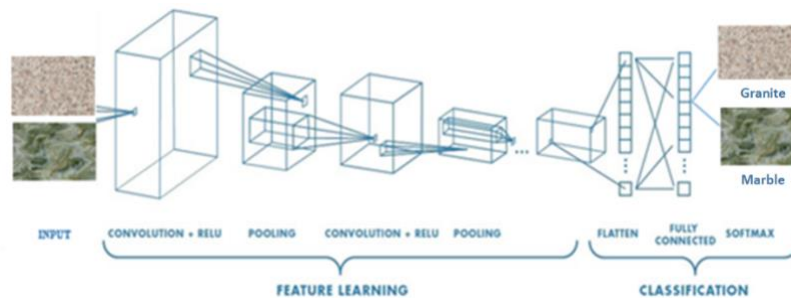


Fig. 1. An illustration of CNN [20].

The Proposed Method

The proposed method uses VGG16 [21] as the model in the feature extraction stage and add some fully connected layers to classify the features into 6 types of tiles. Fig. 2 shows the procedure of the proposed method where the blue lines indicate the training steps, and the orange lines indicate the test steps. The output is category label that represent the type of tiles.

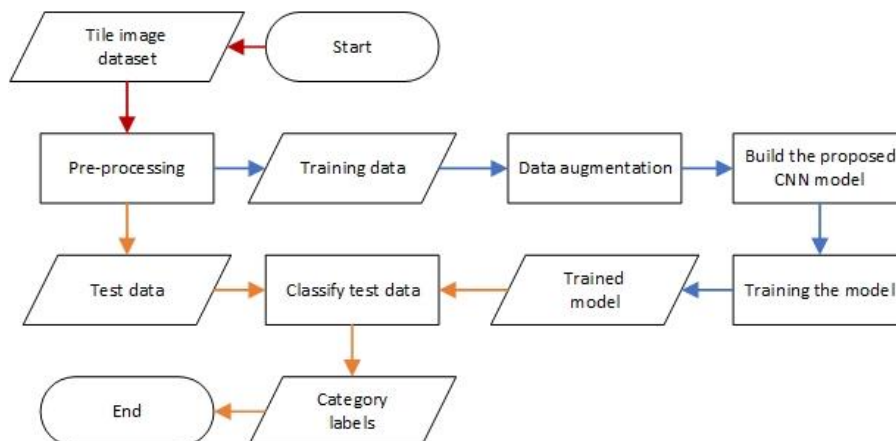


Fig. 2. The procedure of the proposed method.

Based on Fig. 2, the tile images dataset is collected then proceed to the preprocessing step, by cropping the image to get the ROI of tiles object which aims to speed up the computational process. The image data then divided into training data and test data. Training data will become the input in the training step and test data will become the input in the testing step. In the training step, the training data will be split further to generate validation data to assist in the training process. Next is the augmentation step which aims to improve images in the training data in order to get high accuracy results. The proposed model is built using VGG16 model as the feature extractor and fully connected layers as the classifier. The VGG16 architecture is shown in Fig. 3.

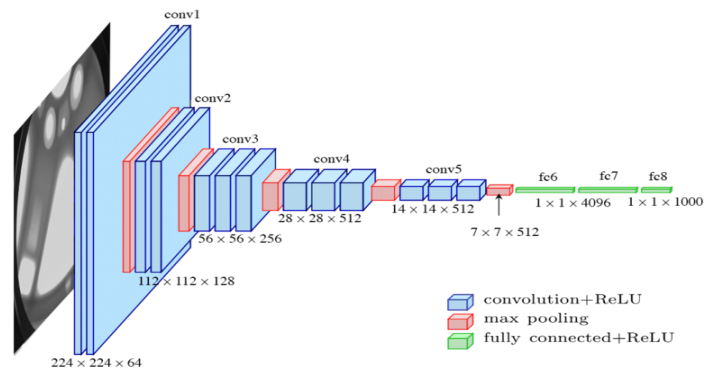


Fig. 3. VGG16 architecture. (image source: [22])

The VGG16 model will be trained from scratch without using the pre-trained weights from ImageNet [23]. After performing average pooling in the last convolutional layer of VGG16, three fully connected layers are used to classify the features into 6 types of tiles. The architecture of the proposed method is shown in Fig. 4.

```
Model: "Ceramic_model"
```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
tf.__operators__.getitem (S1)	(None, 224, 224, 3)	0
tf.nn.bias_add (TFOpLambda)	(None, 224, 224, 3)	0
vgg16 (Functional)	(None, 512)	14714688
dense (Dense)	(None, 4096)	2101248
batch_normalization (BatchNo)	(None, 4096)	16384
activation (Activation)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16781312
batch_normalization_1 (Batch)	(None, 4096)	16384
activation_1 (Activation)	(None, 4096)	0
main_model (Dense)	(None, 6)	24582

```

Total params: 33,654,598
Trainable params: 33,638,214
Non-trainable params: 16,384

```

Fig. 4. The proposed VGG16-based CNN architecture.

Based on Fig. 4, the input layer takes an image with dimensions of (224, 224, 3). After the image is processed at each layer in the VGG16 model, global average pooling is applied, and the features

dimension become 512. The features then further processed by two fully connected layers with 4096 units followed by batch normalization and ReLU activation function. The final fully connected layer classifies the features into 6 category labels using SoftMax activation function.



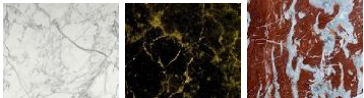


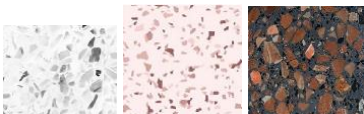
RESULT AND DISCUSSIONS

The proposed method is built using Python, running on Google Collab notebooks with additional TensorFlow, OpenCV, and Scikit-learn library. The VGG16 and other models used in this research are taken from the built-in models provided by TensorFlow library [24].

Dataset

The specification of tiles dataset that used in this research is shown in Table 1. Based on Table 1, there are 6 types of tiles namely granite, limestone, marble, motifs, mosaics, and terrazzo. Each category consists of 57 images and the total is 342 images. The tile images are collected from the internet and taken directly from the local tile shops. Tiles dataset is divided into three namely training data, validation data and test data with each of them consists of 186, 96, and 60 images respectively. The images are resized into 224x224 before trained by the model. Data augmentation is performed to enrich the training data by flipping horizontal, rotation in range of 90 degrees, shear by factor 0.2, and scaling by factor 0.2.

Table 1. Tiles dataset specification.

No	Tiles type	Data type	Qty	Image source	Image samples
1	Granite	jpg	57	Internet and tile shop	
2	Limestone	jpg	57	Internet	
3	Marble	jpg	57	Internet and tile shop	
4	Motif	jpg	57	Internet	
5	Mosaic	jpg	57	Internet and tile shop	
6	Terrazzo	jpg	57	Internet	

Training Results

Before conducting the training process, some hyperparameters need to be configured. The model is trained using Adam optimizer with 0.0001 learning rate and runs in 100 epochs. Fig. 5 shows the result of training process namely accuracy and loss of training data (orange lines) and validation data (blue lines). Accuracy is the value used to determine the level of success of the model that has been made while loss is a measure of the error that has been made with the aim of minimizing it. Based on Fig. 5, the accuracy of the training data reached 100% with 0.0049 of loss while the accuracy of validation data reached 94%.

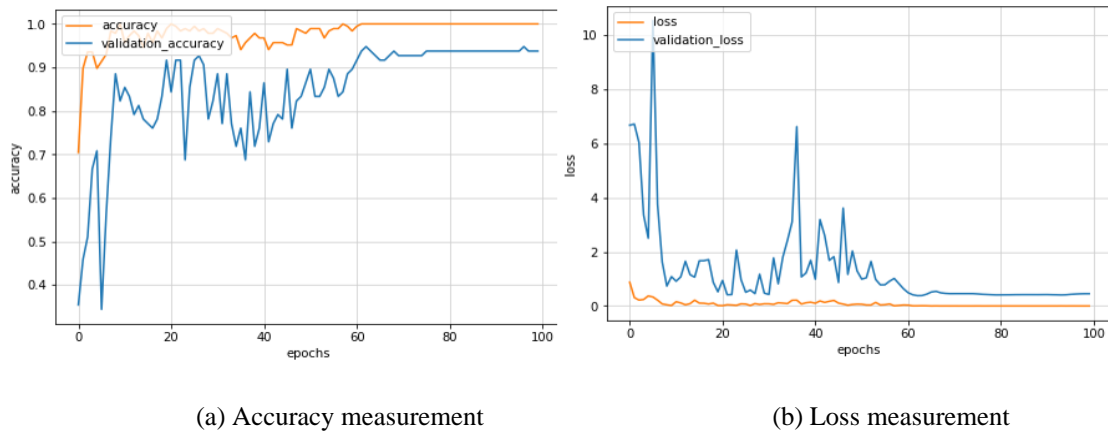


Fig. 5. Accuracy and loss of training data (orange) and validation data (blue).

As the model performed well in validation data, we tested the model on the test data to measure the accuracy of the trained model. The test data consists of 60 images with 10 images for each tile category namely granite, limestone, marble, motifs, mosaics, and terrazzo. Table 2 shows the confusion matrix of prediction results in test data. Based on Table 2, there is only one false positive where a marble is classified as limestone. The model achieves 98.33% of accuracy according to the confusion matrix in Table 2.

Table 2. Confusion matrix of the prediction results.

		Prediction						Total
		Granite	Limestone	Marble	Motive	Mosaic	Terrazzo	
Actual class	Granite	10	0	0	0	0	0	10
	Limestone	0	10	0	0	0	0	10
	Marble	0	1	9	0	0	0	10
	Motive	0	0	0	10	0	0	10
	Mosaic	0	0	0	0	10	0	10
	Terrazzo	0	0	0	0	0	10	10

$$\text{Overall accuracy} = \frac{\text{total correct prediction on class data}}{\text{number of class data}} = \frac{59}{60} = 98.33\%$$

We also perform training and test for several models to compare the proposed CNN which based on VGG16 model with the other CNN models namely ResNet50 [25], InceptionV3 [26], InceptionResNetV2 [27], and MobileNetV2 [28] that provided by TensorFlow library [24]. The other models also used as the feature extractor with the same hyperparameters configuration. The

purpose of this experiment is to know which model produces the best accuracy. The performance evaluation result is shown in Table 3 (we round up the accuracy score). Based on Table 3, although InceptionV3 achieves the highest accuracy on validation data which is 98% but the performance on the test data only achieves 25%. Meanwhile, the proposed VGG16-based CNN model only achieves 94% accuracy on validation data but achieves the highest accuracy on test data which is 98% compared with the other models. Therefore, the performance of the proposed CNN method is superior than the other models in this dataset. Based on the classification result, we can get the usage and the advantages of the corresponding tile from the database to show the detail information to the user.

Table 3. Performance evaluation comparison of several CNN models.

No	Model name	Reference	Epoch	Accuracy	
				Validation data	Test data
1	Proposed VGG16-based CNN model		100	94%	98%
2	ResNet50	[25]	100	97%	33%
3	InceptionV3	[26]	100	98%	25%
4	InceptionResNetV2	[27]	100	97%	18%
5	MobileNetV2	[28]	100	97%	20%

CONCLUSIONS

The proposed VGG16-based CNN model in this study shows a good performance in classifying 6 types of tiles namely granite, limestone, marble, motifs, mosaics, and terrazzo tiles. The best accuracy result produces by the proposed VGG16-based CNN model where in the training process achieves 100%, in the validation process achieves 94% and, in the testing, process achieves 98.33%. From the experiment results, it can be concluded that the application of deep learning using the Convolutional Neural Networks can classify tiles well. The classification result then can be used to get the usage and advantages of the corresponding tiles from the database and shows the detail information to the user. In the future research, it is expected to expand the classification for more types of tiles.

REFERENCES

- [1] J. Framinan, R. Leisten, and R. Ruiz, "A Case Study: Ceramic Tile Production," 2014, pp. 371–395. doi: 10.1007/978-1-4471-6272-8_15.
- [2] A. Ferreira and G. Giraldi, "Convolutional Neural Network approaches to granite tiles classification," *Expert Syst. Appl.*, vol. 84, pp. 1–11, 2017, doi: 10.1016/j.eswa.2017.04.053.
- [3] W. Chen, Q. Sun, J. Wang, J.-J. Dong, and C. Xu, "A novel model based on AdaBoost and deep CNN for vehicle classification," *Ieee Access*, vol. 6, pp. 60445–60455, 2018.
- [4] J. Tristanto, J. Hendryli, and D. Herwindiati, "Classification of batik motifs using Convolutional Neural Networks," 2018.
- [5] A. B. Perdana and A. Prahara, "Face Recognition Using Light-Convolutional Neural Networks Based On Modified Vgg16 Model," in *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*, 2019, pp. 1–4.
- [6] S. Winiarti, A. Prahara, D. P. I. Murinto, and P. Ismi, "Pretrained convolutional neural network for classification of tanning leather image," *Netw. CNN*, vol. 9, no. 1, 2018.
- [7] F. A. Faiz and A. Azhari, "Tanned and Synthetic Leather Classification Based on Images Texture with Convolutional Neural Network," *Knowl. Eng. Data Sci.*, vol. 3, no. 2, Art. no. 2, Dec. 2020, doi: 10.17977/um018v3i22020p77-88.

- [8] G. A. Pamiela and A. Azhari, "Deep Learning on EEG Study Concentration in Pandemic," p. 10, doi: 10.30872/jim.v16i2.6474.
- [9] L. A. M. Pangistu and A. Azhari, "Deep Learning on Game Addiction Detection Based on Electroencephalogram," *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 3, p. 963, Jul. 2021, doi: 10.30865/mib.v5i3.3061.
- [10] W. R. Gould, "The 12 Different Types of Tiles, Explained by Pros," 2019.
- [11] E. V. Sousa, L. A. F. Fernandes, and C. N. Vasconcelos, "ConformalLayers: A non-linear sequential neural network with associative layers," no. 1, pp. 386–393, 2021, doi: 10.1109/sibgrapi54419.2021.00059.
- [12] İ. Karaali and M. Eminağaoğlu, "A convolutional neural network model for marble quality classification," *SN Appl. Sci.*, vol. 2, no. 10, 2020, doi: 10.1007/s42452-020-03520-5.
- [13] A. G. Ouzounis, G. K. Sidiropoulos, G. A. Papakostas, I. T. Sarafis, A. Stamkos, and G. Solakis, "Interpretable deep learning for marble tiles sorting," *Proc. 2nd Int. Conf. Deep Learn. Theory Appl. DeLTA 2021*, no. DeLTA, pp. 101–108, 2021, doi: 10.5220/0010517001010108.
- [14] R. Alamsyah, A. D. Wiranata, and R. -, "Defect Detection of Ceramic Tiles using Median Filtering, Morphological Techniques, Gray Level Co-occurrence Matrix and K-Nearest Neighbor Method," *Sci. Res. J.*, vol. VII, no. IV, pp. 41–45, 2019, doi: 10.31364/scirj/v7.i4.2019.p0419632.
- [15] S. L. Oh, J. Vichesh, E. J. Ciaccio, R. Yuvaraj, and U. R. Acharya, "Deep convolutional neural network model for automated diagnosis of Schizophrenia using EEG signals," *Appl. Sci. Switz.*, vol. 9, no. 14, 2019, doi: 10.3390/app9142870.
- [16] S. Purwar, R. Tripathi, R. Ranjan, and R. Saxena, "Classification of thalassemia patients using a fusion of deep image and clinical features," *Proc. Conflu. 2021 11th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 410–415, 2021, doi: 10.1109/Confluence51648.2021.9377054.
- [17] E. Maggiori, Y. Tarabalka, G. Charpiat, P. Alliez, and S. Member, "Convolutional Neural Networks for Large-Scale Remote Sensing Image Classification IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING 1 Convolutional Neural Networks for Large-Scale Remote Sensing Image Classification," *Ieee Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, 2017.
- [18] Q. Gao, S. Lim, and X. Jia, "Hyperspectral image classification using convolutional neural networks and multiple feature learning," *Remote Sens.*, vol. 10, no. 2, 2018, doi: 10.3390/rs10020299.
- [19] S. Kilicarslan, M. Celik, and Ş. Sahin, "Hybrid models based on genetic algorithm and deep learning algorithms for nutritional Anemia disease classification," *Biomed. Signal Process. Control*, vol. 63, 2021, doi: 10.1016/j.bspc.2020.102231.
- [20] M. Ather, I. Hussain, B. Khan, Z. Wang, and G. Song, "Automatic recognition and classification of granite tiles using convolutional neural networks (CNN)," *ACM Int. Conf. Proceeding Ser.*, pp. 193–197, 2019, doi: 10.1145/3369114.3369124.
- [21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [22] M. Ferguson, R. ak, Y.-T. Lee, and K. Law, "Automatic localization of casting defects with convolutional neural networks," 2017, pp. 1726–1735. doi: 10.1109/BigData.2017.8258115.
- [23] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.
- [24] TensorFlow, "Tensorflow Keras Model Applications."
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

- [27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.” 2016.
- [28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.