Evaluating on the Transfer Learning of CNN Architectures to a Construction Material Image Classification Task

Supaporn Bunrit, Nittaya Kerdprasop, and Kittisak Kerdprasop

***Abstract*—Various sub-tasks on modern construction management system require automatic or semi-automatic processes in handling the operation inside. Especially for construction progress monitoring task, the automatic process in classifying the difference of each construction material from an image is necessary in the preliminary stage. The more the preciseness in automatic classifying, the more the exactness in assessment of each material had been used. Subsequently, the progress of the construction can be evaluated with the highest degree of reliability. As a result, classification of construction material images is very essential process for automatic progress monitoring. Whereas, the similarities in material image appearances are the major classifying challenges. All most all existing related works have been studied based on hand-designed features of which the classified accuracy still not much appreciated from different studied datasets. In our work, automatic feature extracted method from the prominent technique in deep learning, convolution neural network (CNN), is proposed. The pre-trained CNN architectures of AlexNet and GoogleNet are adopt with the task of construction material images classification in the concept of transfer learning. Both of fixed feature extractor and fine-tuning schemes of transfer learning are technically implemented and evaluated. Analyzing results from the two pre-trained architectures expose very impressive and interesting circumstances to the studied dataset. Entirely, fine-tuning scheme of GoogleNet reveals the highest classification result by 95.50 percent of accuracy.**

***Index Terms*—Convolution neural network (CNN), deep learning, transfer learning, construction material, image classification.**

I. Introduction

At present, digital image processing and computer vision are the progressive research directions in architecture, engineering, construction, and facilities management (AEC/FM) [1]. Such directions when incorporate to the advancement of machine learning techniques can be solved for many tentative applications. In a field of construction management, automatic or semi-automatic systems for various sub-tasks are needed. Particularly for construction progress monitoring task, automatic classifying the difference between the construction materials is essential in the preliminary procedure. Where source data of construction materials must be acquired by camera in a form of image or video cause, other technologies could not indicate the difference among materials [2]. The useful information from image or video, therefore, must be extracted and identified by some efficient methods. As a result, in an application of construction progress monitoring, the classification of construction material from images must be as precise as possible. In order that the subsequent steps in evaluating the progress of the construction can perform with the highest degree of reliability.

In literatures, the methods involved construction material image classifications were studied based on hand-designed features. Where the prominent algorithms in digital image processing or computer vision were applied to extract the expected features and the suitable classifier was selected to classify such features. Therefore, the classification accuracy depends on manual selection of the feature-extracted algorithm. For our proposed work, automatic feature extraction method by a novel CNN in deep learning technique is employed. By the way, CNN based methods can separate into two scenarios, which are learning from scratch and transfer learning. In this work, we explore on the transfer learning. Two of the difference pre-trained architectures which are AlexNet [3] and GoogleNet [4] are technically transferred to a construction material image classification task. Both fixed feature extractor and fine-tuning schemes of transfer learning are evaluated from such two architectures.

II. Literature Reviews

Material images classification method initiatively studied by Brilakis *et al*., [5] in an application of material image retrieval. A series of content-based filters were employed in such work to decompose an image into color, texture, and structure features. They used knowledge database to compare the computed feature signature of each cluster after dividing an image into cluster region. The interval of each feature signature was done by threshold and comparing was measured by Euclidean distance. Zhu and Brilakis [6] also considered machine-learning techniques for identifying concrete material regions. Firstly, segmentation was applied to divide the construction site image into regions. Then, visual features from color and texture were used to classify by support vector machine (SVM) against artificial neural network (ANN). Experiment revealed the performance from ANN was better than SVM of which the average of precision and recall were around 80%. In 2016, Rashidi *et al*., [1], proposed an analogy between various machine-learning techniques for detecting construction material of building. The studied materials were concrete, red brick, and OSB (Oriented Strand Board). The comparison classifiers were multi-layer perceptron (MLP), radial basis function (RBF), and SVM. Where RGB histogram, HSV histogram, and histogram of dominant edges were extracted as the features.

Manuscript received August 25, 2018; revised November 1, 2018. This work was supported by grants from Suranaree University of Technology (SUT), Thailand.

The authors are with the School of Computer Engineering, SUT, Thailand (corresponding author: S. Bunrit; Tel.: +66944961244; e-mail: sbunrit@sut.ac.th, nittaya@sut.ac.th, kerdpras@sut.ac.th).

Experiments conducted based on two-class of problem classification; target and non-target class of materials. The best accuracy was from SVM with RBF kernel.

Son *et al*., [7] explored the performance of six classifiers and the potential of ensemble classifies on three materials, which are concrete, steel, and wood. Voting based ensemble was created by six different classifiers which are SVM, ANN, Commercial version 4.5 (C4.5), Naïve Bayes (NB), Logistic regression (LR), and k-Nearest neighbors (KNN). Three values from HSI color space are used as features. The accuracy, precision, sensitivity, and average score values were measuring and comparing. The ensemble classifier was significantly better than each single classifier. In 2014, Dimitrov and Golparvar-Fard [2] technically proposed a bag of words (BoW) pipeline for forming statistical distributions of materials and multiples of binary SVM were used as the classifiers. The material appearances were modeled by joint probability distribution of response from a filter bank and principle HSV color values. In this work, they also proposed the prototype of the construction material library and the validation metrics. 3D geometry information of materials was investigated incorporated to 2D features in a work of DeGol *et al*., [8]. Considering features of 3D geometries were surface normal, camera intrinsic, and extrinsic parameters. 2D features were fisher vector, HSV color, and CNN feature from pre-trained VGG-M network. A one vs. all SVM scheme was used as the classifier. New dataset, which provide both images and geometry data, had been public in this work. They experimented on various combinations of 2D and 3D features. The results revealed the combination of surface normal, fisher vector, and CNN feature got the highest accuracy of which 73.84%. Whereas, when considered only 2D features the best accuracy was 68.92% from fisher vector incorporated to CNN feature.

All mentioned existing methods for the classification task of material image were proposed based on hand-designed features. That means the specific ways of the extracted features must be identified before the classification process. For such methods, none of the automatic feature extracted method such as deep learning technique has been directly studied. Although DeGol *et al*., [8] used CNN feature in their work, such feature only explored incorporated to other features in order to study about the important of 3D geometry. They did not focus the studied in particular to CNN network applying for construction material dataset. For our proposed work, as a result, a new notable scenario of CNN based method which is transfer learning is applied and evaluated for material image classification task. Two of pre-trained architectures trained on ImageNet dataset (based task) are explored in order to evaluate that if there are two architectures pre-trained on the same based task, which one is suitable to our task specific (construction material dataset). We select the two distinct pre-trained architectures of which both differences in deep and in their layer details; AlexNet and GoogleNet.

III. Theories

A. Convolution Neural Networks (CNNs)

CNN is a particular neural network model of which the convolution is employed as a key operation in a network. The network for classification task can separate into feature learning process and classification process as shows in Fig. 1. In feature learning process, three principles of stage are used in order to learn and extract the features from input, which are convolution stage follow by nonlinearity stage using rectified linear unit function (ReLU) and subsampling stage name by pooling. Such many of these stages are consecutively used as layer-by-layer aimed at automatically extracting the features in deep. Therefore, each of stage may views as each of a layer in the network. The features learned by feature learning process will further send forward to the classification process where the fully connected (FC) manner as the traditional multilayer perceptron (MLP) is used. Finally, softmax function is employed for a layer before output layer in order to gain the output in a probability fashion. The layer details of a network shows in Fig. 1 is an architecture of AlexNet that won on ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012. It views as an architecture that consists of eight weighted layers (when count only the layers that has weights to be adapted). If count all of the detailed layers, it can separate into 25 consecutive layers shows on the right side of Fig. 1.

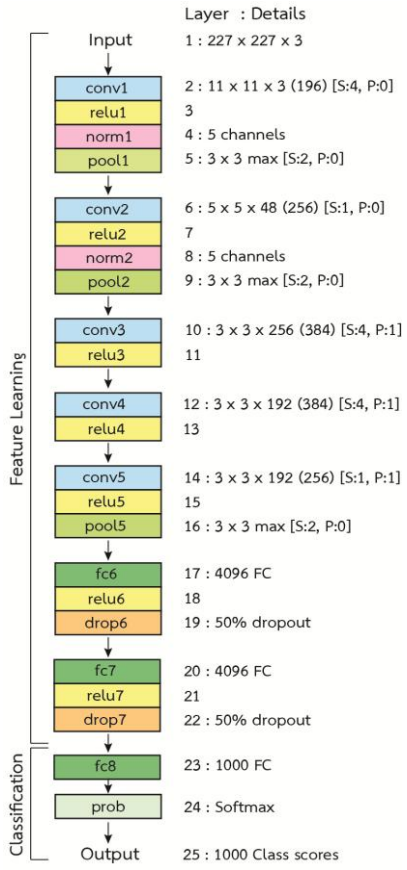


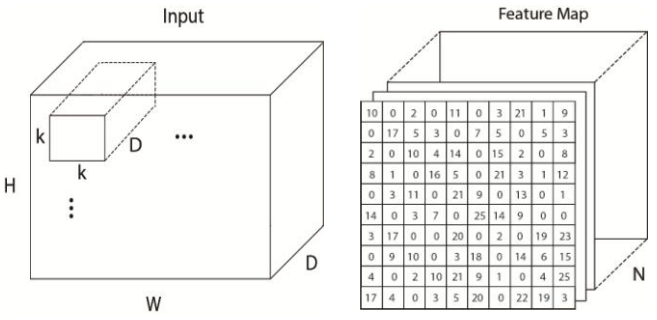
Fig. 1. CNN architecture details. Example of AlexNet [3] trained on ImageNet dataset.

Each convolution layer of CNN use many filters of size $k \times k \times D$ to convolute with the incoming input in the formed of 2D convolution with 3D input as shown in Fig. 2(a). Such filters are used once at a time to convolute with the 3D input of size $W \times H \times D$ in a form of sliding windows. Therefore, an input image of RGB color will has D equal to three from three-color channel of R, G, and B. The convolution result from one filter is one of the feature map output. As a result,

when *N* filters are employed, total output from a convolution stage is the stacks of *N* feature maps as depict Fig. 2(b). Actually the output from convolution may result in negative values, in Fig. 2(b) shows the values when ReLU function is already applied. Because convolution is a linear operation, the results from the convolution stage of CNN network will pass through a non-linear ReLU function in order to extract the non-linear property of the features. ReLU function is shown in (1). The function converts all the negative values to zero where keeps the others as the original.

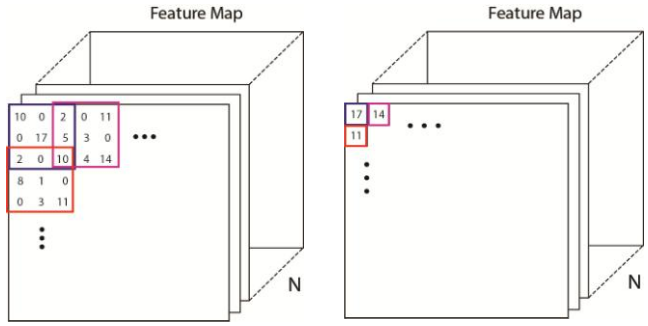
$$\phi(x) = \max(0, x) \tag{1}$$

Another key operation in CNN is pooling. These stage uses for subsampling to the previous stage data. After pooling, the dimension of width (W) and height (H) of the data will decrease. Fig. 2(c) shows an example of pooling operation when subsampling of the data by windows size of 3×3 and stride (slide to the right or to the bottom) by two positions. Its mean only one value from nine values is selected as the subsampling value. Thereby, average pooling, max pooling, or any others pooling types can be used to select one subsampling value from such nine values. Fig. 2(d) is the results after max pooling is applied to the 3×3 windows of the region marked in Fig. 2(c). After pooling stage, as a result, the width and height of the feature map are smaller.



(a). Tensor input and filters (weight)

(b). Example of feature map output after applies "conv" with N filters follows by "relu".



(c). 3×3 pooling with stride 2 (S:2).

(d). Result after applies "pool" with 3×3 max to location in figure c.

Fig. 2. The operations in major layers of CNN.

In feature learning process, many layers consisting of the convolution (conv), nonlinearity by ReLU (relu), and pooling (pool) stages are consecutively arranged to form a network. Some other stages may include such as in AlexNet shown in Fig. 1, the stage of normalization (norn) and dropout (drop) are also used.

For the classification process, some of fully connected layers (fc) are used in order to classify the extracted features from the last layer of the feature learning process. For the last layer of CNN before output, softmax function is employed to transform the output of the network to be the values in term of probability. Softmax function shows in (2).

$$S(y_i) = \frac{\exp^{y_i}}{\sum_{j=1}^{J} \exp^{y_j}} \tag{2}$$

where

(y_i) : the softmax result of each y_i,

y_i : an output of each i,

$\exp^{y_i}$: the exponential value of y_i,

and j : the component of vector y

B. CNN Based Methods

When we want to apply CNN network to our application domain we can do in two different CNN based methods show in Fig. 3. The first method knows in term "learning from scratch" when the CNN network that appropriated to the studied dataset (task specific dataset) are generated and fully train on such dataset. The second method is the transferring of knowledge (in term of weight and bias values) from some of the pre-trained architectures trained by other dataset (based task). Such knowledge from pre-trained architecture is transferred to the task specific dataset. For this way knows in the term "transfer learning" [9]. Transfer learning of knowledge can also apply by two schemes of which fixed feature extractor and fine-tuning depicted in Fig. 3. Fixed feature extractor directly uses the pre-trained weights and bias transferred to a task specific by no need to retain the network. Opposite to the fine-tuning, the network must be retrain on some parts of a network using a task specific dataset with weights and bias initialized from transferring pre-trained weights and bias.

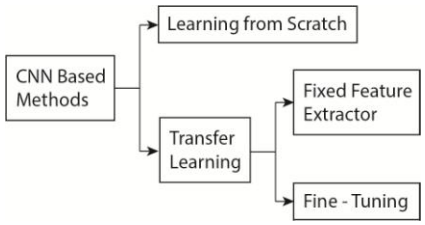


Fig. 3. CNN based methods.

C. CNN Pre-trained Architectures

The name of CNN has been well known since 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC 2012) when CNN Architecture of AlexNet [3] by Krizhevsky *et al.*, won the competition. Since then, year-by-year different deep learning architectures of CNN still were the winner of ILSVRC. There were ZF Net [10] in 2013, GoogleNet in 2014 [4], and ResNet in 2015 [11], respectively. After that, deeper and deeper architectures are proposed by the combination of GoogleNet and ResNet concepts. It has been known in machine learning community that fully trains of CNN network to a task specific dataset needs huge of computer resources and take time. Most of all, dataset size must effect to the performance. By the reasons, transfer of learning approach has been come up and many of well-known CNN pre-trained architectures are public.

In this research, two of pre-trained architectures trained on

based task of ImageNet dataset are explored. We select the two distinct pre-trained architectures of which both differences in deep and in their layer details; AlexNet and GoogleNet. Such architectures are public as the pre-trained networks with natural images in ImageNet dataset. Both were per-trained by around 1.2M images of 1000 classes of everyday used images. Architecture detailed of each explained as follow:

AlexNet

The architecture details of AlexNet already shown in Fig. 1. In total, it used five of convolution (conv) layers, two of max pooling (pool) layers, two of normalization (norm) layers, and three of fully connected (fc) layers. Nowadays, pre-trained architecture of AlexNet using ImageNet dataset is public and transfers to many application domains.

GoogleNet

In 2014, Szegegy *et al*., from Google research term developed architecture of CNN shown in Fig. 4 for ILSVRC 2014 and won the competition. The network quite deeper than AlexNet of which view as consisting of 22 weighted layers. They proposed a network under an improvement on the calculation resources. The efficient of a network came from both wider and deeper by incorporating nine modules of "inception module" on some parts of a network as shows in Fig. 4(a). Details of each inception module are in Fig. 4(b). Only small filter size of 1×1, 3×3, and 5×5 are used in the module. Each block in a module can do in parallel and the results from all blocks are concatenated to be inception module output send to the next layer.

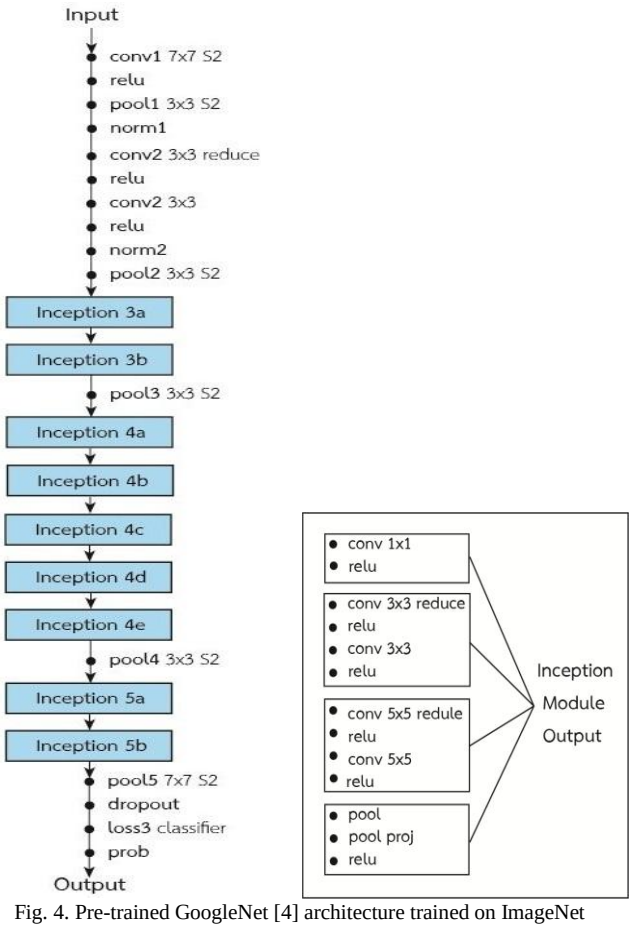


Fig. 4. Pre-trained GoogleNet [4] architecture trained on ImageNet dataset.

If the architecture of GoogleNet is pictured as AlexNet in Fig. 1 it will consist of around 144 detailed layers. Therefore, its architecture is more complex both in deep and in details.

IV. Proposed Method

A. Contributions

Our research is the first work that directly explore about transferring the pre-trained CNN architecture to a task specific on construction material images classification. Under our study, transfer-learning scenario of CNN based method is technically adopts in order to evaluate as follow:

1) If there are two of CNN architectures pre-trained on the same based-task, which one is the most suitable for transferring to our task specific?
2) For fixed featured extractor scheme, related works (from many task specifics) always explore just the feature from the last layer of feature learning. Is it can reveal any interesting result if we explore and evaluate on other learned features from the intermediated layers?
3) For fine-tuning scheme, what are the suitable important parameters selected for each pre-trained architecture? Such parameters must use to retrain in the fine-tuning process, e.g. learning rate, size of mini-batch, and number of epoch.
4) Can the transfer learning from the explored pre-trained CNN architectures achieve an attractive result to our task specific?

Under the evaluation according to our contribution, the two architectures we select to study are AlexNet depicted in Fig. 1 and GoogleNet shown in Fig. 4. These two distinct pre-trained architectures are difference in their deep and their layer details. Both pre-trained on the same based-task, which is consisting of around 1.2M images of 1000 classes from ImageNet dataset.

B. Transfer Learning by Fixed Feature Extractor

Fixed feature extractor scheme is a method that the transferred weights and bias from the pre-trained architecture are directly used for the classification process by no need to retrain the network with the training set of the task specific data set. That mean, the task specific data can directly transform to the pre-trained features and any classifier can further classify such transformed features related to the target class of the task specific dataset.

Our study task specific dataset of construction material image, there consist three classes of materials, which are brick, concrete, and wood. In order to evaluate the performance of fixed feature extractor, we use support vector machine (SVM) as a classifier. According to the contribution number two mentions previously, instead of explore only the last layer of feature learning process as done on most works, we investigate on many of intermediated featured layers.

C. Transfer Learning by Fine-Tuning

In order to fine-tuning the pre-trained architecture to a task specific dataset, we can implement by retraining some part of the pre-trained network with the training set of the task specified dataset. Fine-tuning process use the following steps:

1) Replace the output layer of a pre-trained architecture to match the number of target class exist on the studied

dataset. Therefore, fine-tuning of AlexNet and GoogleNet to our studied dataset, output layer in Fig. 1 and Fig. 4 must be changed from 1000 class scores to 3 classes scores (because our studied dataset has 3 target classes). As a result, our fine-tuning network has only three output neurons.

2) Set the initial values of all weights and bias for part of the fine-tuning network with the transferred pre-trained weights and bias.
3) Set the training parameters of CNN, which are learning rate, mini-batch size, number of epoch or number of iteration to be learned. This may include the momentum and regularization parameters.
4) Train the fine-tuning network with the training set of the task specified dataset.
5) Evaluate the fine-tuning performance by the testing set of the task specified dataset.

The suitable important parameters used in step 3 above come from the stochastic gradient descent (SGD) optimization algorithm used in CNN learning. Equation 3 [12] expresses the empirical loss with regularization term we want to minimize. Such minimization is done by the training samples $(X_i, Y_i)_{1 \le i \le n}$ to estimate the parameters θ (all weights and bias).

$$L_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(f(X_i, \theta), Y_i) + \lambda\Omega(\theta) \quad (3)$$

where,

$L_n(\theta)$: the empirical loss,

$l(f(X_i, \theta), Y_i)$: the loss function,

and $\lambda\Omega(\theta)$: the regularization term

In order to minimize $L_n(\theta)$ in (3), stochastic gradient descent algorithm is used. Such algorithm performs by adapting the parameters θ as (4).

$$\theta^{k+1} = \theta^k - \mathcal{E}\frac{1}{m}\sum_{i \in B}[\nabla_\theta l(f(X_i, \theta), Y_i) + \lambda\nabla_\theta\Omega(\theta)] \quad (4)$$

where, k: iteration number, $\mathcal{E}$: leaning rate, m : mini-batch size, B : samples in each mini-batch, and ∇_θ: gradient of θ.

According to (4), when train CNN network, the gradient for the loss function do not compute at each iteration, but only on a set B. Where size of B equal to mini-batch size (m). This procedure called mini-batch learning, which is an approach always use in deep learning algorithms including CNN. Therefore, the important parameter in (4) needed to set is leaning rate, mini-batch size, and total numbers of epoch, where one epoch counted for a pass of all samples to the network. In our work, these parameters are observed based on empirical experiments.

V. Experimental Results and Discussions

A. Data Sets

Task specific dataset use in this study consists of three prominent classes of material image, which are brick, wood, and concrete. There are parts of the public images in a work of DeGol *et al*., [8]. Examples of some images in each class show in Fig. 5. The left column is brick, the middle is concrete, and the right is wood, respectively. All images are 100×100 pixels resolution. The training set consists of 400 images of per class and the testing set is 200 images per class.

brick concrete wood

Fig. 5. Examples of some images in each class.

B. Experimental Results

1) Fixed feature extractor

Following our contributions mentioned in part *A* of Section IV, we set the experiments related to such contributions. Table I shows the result of fixed feature extractor from AlexNet when we investigate on many of intermediated layers. Most research of many application tasks always used the feature from layer "fc7" marked by underline. For our work, feature from such layer reveal a bit poor result compare to the others. The highest classification accuracy is form the feature in layer "pool5" with 91.83% of accuracy labeled by bold font of Table I.

TABLE I: Classification Accuracy Using Fixed Feature Extractor of **AlexNet** from Difference Layers (Layer Name Refer to Fig. 1)

Extracted Feature from Layer	Accuracy (%)	Extracted Feature from Layer	Accuracy (%)
conv4	70.83	fc6	91.17
relu4	87.33	drop6	91.17
conv5	91.00	fc7	89.67
relu5	91.50	relu7	90.67
pool5	**91.83**	drop7	90.67
fc6	91.67	fc8	87.50

Table II are the results from GoogleNet when experiment on fixed feature extractor scheme. Most research used the feature from layer "loss3" marked by underline. Nevertheless, for our task, such layer exposes very poor result compare to the layer "Inception5a" marked by bold font. Pre-trained feature from Inception5a layer get 92.33% of classification accuracy.

TABLE II: Classification Accuracy Using Fixed Feature Extractor of **GoogleNet** from Difference Layers (Layer Name Refer to Fig. 4)

Extracted Feature from Layer	Accuracy (%)	Extracted Feature from Layer	Accuracy (%)
Inception3a	79.17	Inception4e	92.00
Inception3a	79.67	pool4	91.33
pool3	87.33	**Inception5a**	**92.33**
Inception4a	86.33	Inception5b	90.17
Inception4b	89.17	pool5	91.17
Inception4c	89.50	dropout	91.17
Inception4d	92.00	loss3	86.17

2) Fine-tuning

The experiments on fine-tuning scheme are conducted based on the parameters setting for each architecture according to Table III. Such parameters are observed based on an empirical experiments follow a work in [13]. Highest classification accuracy from the test set data are got when use parameters shown in Table III. Besides, both architectures we set the momentum term to be 0.9 and the regularization term to be 0.5. As such, we are fine-tuning the network by the stochastic gradient descent with momentum (SGDM) algorithm.

TABLE III: PARAMETERS SETTING IN FINE-TUNING PROCESS OF BOTH ARCHITECTURES

Architecture	Learning rate	Mini-batch size	No. of epoch
AlexNet	0.0001	5	20
GoogleNet	0.0001	5	15

After fine-tuning, the performance of both AlexNet and GoogleNet are improved when compared to the fixed feature extractor scheme. Fig. 6 shows the results from AlexNet by using confusion matrix. In total, the test set consists of 600 images from three classes. There are 200 image or 33.33% for each class. Overall accuracy from AlexNet fine-tuning is 94.5% marked by bold font. When consider on per class classification, it can classify concrete class with the highest accuracy of which 97.5% that label by italic bold font.

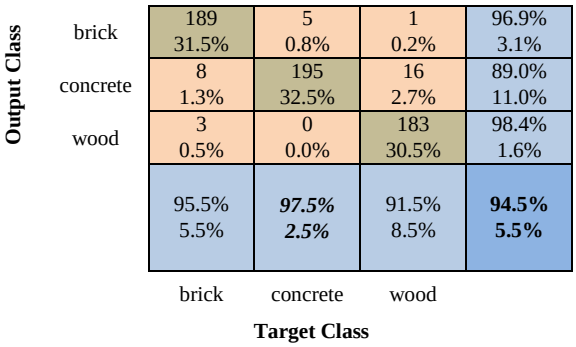


Fig. 6. Confusion matrix of the classification results from AlexNet with fine-tuning.

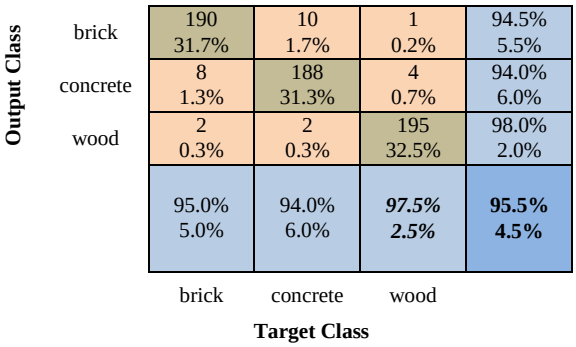


Fig. 7. Confusion matrix of the classification results from GoogleNet with fine-tuning.

Fig. 7 is the confusion matrix results from GoogleNet fine-tuning. Overall accuracy is 95.5% marked as bold font. For per class classification, class of wood can classify with the highest accuracy of which 97.5% represent as italic bold font.

Table IV shows the overall of the classification results from both schemes of transfer learning and depicts to compare for both architectures. Entirely, fine-tuning scheme of GoogleNet exposes the best classification accuracy of which 95.5%. After fine-tuning, the performance from both architectures are improved. Where, the performance from GoogleNet improves higher than AlexNet around 0.5%.

TABLE IV: OVERALL TRANSFER LEARNING RESULTS FROM BOTH PRE-TRAINED ARCHITECTURES

Architecture	Accuracy	Improvement after fine-tuning
AlexNet		
Fixed Featured Extractor	91.83	NA
Fine-tuning	94.50	2.67%
GoogleNet		
Fixed Featured Extractor	92.33	NA
Fine-tuning	**95.50**	3.17%

C. Evaluations and Discussions

When transfer-learning scenario is adopt to a task specific on construction image classification task it exhibit very interesting circumstances to the studied dataset. First, it achieves very attractive result from 95.5% of the classification accuracy by fine-tuning GoogleNet. Second, for fixed featured extractor scheme, when we investigate on the transferred features from the intermediated layers, the results from such some layers are higher than the layer always used by many existing applications. Third, based on our empirical experiment on the parameters setting for the fine-tuning process, the most performance-affected parameter is the learning rate. These means, if we set the learning rate to 0.01, the performance is much worse than 0.0001 shown in Table III. Finally, from the confusion matrix in Fig. 6, it reveals AlexNet can classify the best for concrete class. While, in Fig. 7, GoogleNet can do the best for class of wood. In addition, both architectures can do quite the same for a class of brick. By this result, it discloses us for the further study whether we can use both the extracted features from both architectures in a combination way such as ensemble or any others for the future work.

VI. CONCLUSION

In this work, a new notable scenario of CNN based method by transfer learning is applied and evaluated for construction material image classification task. Two of pre-trained architectures trained on based task of ImageNet dataset, which are AlexNet and GoogleNet are explored. Both of fixed feature extractor and fine-tuning schemes of transfer learning are technically implemented and evaluated. Analyzing results from the two pre-trained architectures expose very impressive and interesting circumstances to the studied dataset. Best of all, fine-tuning scheme of GoogleNet reveals the highest classification result by 95.50 percent of accuracy.

REFERENCES

[1] A. Rashidi, M. H. Sigari, M. Maghiar, and D. Citrin, "An analogy between various machine-learning techniques for detecting construction materials in digital images," *KSCE Journal of Civil Engineering*, vol. 20, no. 4, pp. 1178-1188, 2016.

[2] A. Dimitrov and M. Golparvar-Fard, "Vision-based material recognition for automated monitoring of construction progress and

generating building information modeling from unordered site image collection," *Advanced Engineering Informatics*, vol. 28, pp. 37-49, 2014.
[3] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *NIPS 2012*, pp. 1106-1114, 2012.
[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CVPR 2014*, 2014.
[5] I. K. Brilakis, L. Soibelman, and Y. Shinagawa, "Construction site image retrieval based on material cluster recognition," *Advanced Engineering Informatics*, vol. 20, pp. 443-452, 2006
[6] Z. Zhu and I. Brilakis, "Concrete column recognition in images and videos," *Journal of Computing in Civil Engineering*, vol. 24, no. 6, pp. 478-487, 2010.
[7] H. Son, C. Kim, N. Hwang, C. Kim, and Y. Kang, "Classification of major construction materials in construction environments using ensemble classifiers," *Advanced Engineering Informatics*, vol. 28, no. 1, pp. 1-10, 2014.
[8] J. DeGol, M. Golparvar-Fard, and D. Hoiem, "Geometry-informed material recognition," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1554-1562.
[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[10] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *ECCV 2013*, pp. 818-833, 2013.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Computer Vision and Pattern Recognition*, 2015.
[12] Neural networks and introduction to deep learning. [Online]. Available: https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning.pdf
[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural network?" *NIP 2014*, 2014.

S. Bunrit is a lecturer with computer engineering school, SUT. She received her bachelor degree in science (mathematics) from Kasetsart University, Thailand, in 1997, the master degree in science (computer science) from Chulalongkorn University, Thailand, in 2001. Her research of interest includes artificial neural network, deep learning, machine learning, digital image processing, computer vision, and time series analysis.

N. Kerdprasop is an associate professor with computer engineering school, SUT. She received her bachelor degree in Radiation Techniques from Mahidol University, Thailand, in 1985, the master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and the doctoral degree in computer science from Nova Southeastern University, U.S.A, in 1999. Her research of interest includes data mining, artificial intelligence, and intelligent databases.

K. Kerdprasop is an associate professor and chair of the School of Computer Engineering, SUT. He received his bachelor degree in mathematics from Srinakarinwirot University, Thailand, in 1986, the master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and the doctoral degree in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes data mining, artificial intelligence, computational statistics.