Improving the Representation of CNN Based Features by Autoencoder for a Task of Construction Material Image Classification

S. Bunrit, N. Kerdprasop, and K. Kerdprasop
Suranaree University of Technology, Thailand
Email: sbunrit@sut.ac.th

***Abstract*—Deep learning based model named Convolution Neural Network (CNN) has been extensively employed by diversified applications concerned images or videos data. Because training a specific CNN model for an application task consumes enormous machine resources and need many of the training data, consequently pre-trained models of CNN have been broadly used as the transfer-learning scenario. By the scenario, features had been learned from a pre-trained model by one source task can be proficiently sent further to another specific task in a concept of knowledge transferring. As a result, a task specific can be directly employed such pre-trained features or further train more by setting the pre-trained features as a starting point. Thereby, it takes not much time and can improve the performance from many referenced works. In this work, with a task specific on construction material images classification, we investigate on the transfer learning of GoogleNet and ResNet101 that pre-trained on ImageNet dataset (source task). By applying both of the transfer-learning schemes, they reveal quite satisfied results. The best for GoogleNet, it gets 95.50 percent of the classification accuracy by fine-tuning scheme. Where, for ResNet101, the best is of 95.00 percent by using fixed feature extractor scheme. Nevertheless, after the learning based representation methods are further employed on top of the transferred features, they expose more appeal results. By Autoencoder based representation method reveals the performance can improve more than PCA (Principal Component Analysis) in all cases. Especially, when the fixed feature extractor of ResNet101 is used as the input to Autoencoder, the classified result can be improved up to 97.83%. It can be inferred, just applying Autoencoder on top of the pre-trained transferred features, the performance can be improved by we have no need to fine-tune the complex pre-trained model.**

***Index Terms*—Convolution Neural Network (CNN), transfer learning, Autoencoder, construction material, image classification**

I. Introduction

Since the emerging of deep learning, Convolution Neural Network (CNN) based learning has been extensively employed by diversified applications. Especially, for the tasks concerned images or videos data. Due to the constructing and learning of a specific CNN model for an application task consumes enormous machine resources and need many of the training data, consequently pre-trained models of CNN have been published and appreciation by many application domains. The features had been learned from a pre-trained model by one source task can be proficiently sent further to another specific task in a concept of transfer learning. By transfer learning, a task specific can be directly employed such pre-trained transfer features or further train more by setting the pre-trained transfer features as a starting point. Thereby, it takes not much time and can improve the performance from many referenced works.

Transfer learning of CNN model can be applied by two schemes, which are fixed feature extractor and fine-tuning. Fixed feature extractor directly transfers pre-trained features to a task specific by just project (activate) the task specified data to such features. Another one popular scheme is fine-tuning. It means the pre-trained transfer features from a source task are fine-tuned to a task specific by training more with a task specific dataset. The result features after retrain are then utilize. Naturally, fixed featured extractor can be process faster than fine-tuning, especially when the pre-trained model is very deep. The deeper of the model, the longer of the fine-tune process. In addition, fine-tuning process need to set many of hyper-parameters. Searching for such suitable and optimal hyper-parameters also take much of time and complex.

In this research, aim at looking for the best performance in construction material images classification task, the novel suitable approaches are then explored. Previous works concerned construction material image classifications were studied based on hand-designed features, of which the outstanding algorithms in image analysis were applied to extract the features and then some classifiers were selected to classify such features. Therefore, the classification accuracy depends on manual selection of the feature-extracted algorithm. In our study, the state of the art approach based on the transfer learning of CNN pre-trained models/architectures is investigated. A set of construction material images is act as a task specific dataset in the transfer-learning scenario. The two selected architectures are GoogleNet [1] and ResNet101 [2] pre-trained on a source task of ImageNet dataset. These two architectures are differences both in deep and in detailed layers.

Manuscript received April 17, 2020; revised September 23, 2020.

doi: 10.12720/jait.11.4.192-199

After the preliminary experiments conduct just on the transfer-learning scenario, we encounter the cumbersome process in fine-tuning scheme. Such fine-tuning process always takes so long time and it is complicated in searching for the optimal hyper-parameters. Especially for ResNet101 which consists of up to 101 weighted layers in deep. Instead of wasting too much of time for fine-tuning, feature learning based representation methods of Autoencoder and PCA are considered in our work. Such two representation methods are applied on top of the transferred pre-trained features. By Autoencoder, the representation features learned from CNN pre-trained based features can improve the performance more than PCA in all cases.

II. Related Works

Many of construction management tasks can be supported by technology progress in computer and internet incorporate to the data acquisition technology. Surveying of the data acquisition technologies used in the construction management applications by Chen *et al.*, [3] indicated laser scanning was used by the most of surveying applications, following by RFID and digital camera, respectively. For construction management tasks involved details of the construction material, acquisition information for the difference of each material must be done solely by camera because information from laser scanning could not indicate the difference among materials [4]. Therefore, digital image processing and computer vision, at this moment, are the progressive research direction in Architecture, Engineering, Construction, and Facilities Management (AEC/FM) [5].

Concerning the construction material classification, in literature Brilakis *et al.*, [6] imitatively explored the method for material images classification in an application of material image retrieval. They employed a series of content-based filters to decompose an image into color, texture, and structure features. Knowledge database was created and used for comparing the feature signature of each cluster when and an image was divided into cluster region. The interval of each feature signature was done by threshold and the comparing was measured by Euclidean distance. Machine learning techniques were considered in a work of Zhu and Brilakis [7] 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%. Rashidi *et al.*, [5] investigated 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 studied 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. 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.

The potential of ensemble classifies were explored by Son *et al.*, [8] They explored the performance of six classifiers 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). Features used are three values from HSI color space. The accuracy, precision, sensitivity, and average score values were measuring and comparing. The ensemble classifier was significantly better than each single classifier. Dimitrov and Golparvar-Fard [4] 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. They also proposed the prototype of the construction material library and the validation metrics. In a work of DeGol *et al.*, [9] 3D geometry information of materials was investigated incorporated to 2D features. The considered features of 3D geometries were surface normal, camera intrinsic, and extrinsic parameters. The 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. When only 2D features were considered, the best accuracy was from fisher vector incorporated to CNN feature.

Related works on construction material images classification were studied based on hand-designed features. Whereas, the specific ways of the extracted features must be identified before the classification process and the classification accuracy depends on manual selection of the feature-extracted algorithm. None of the automatic feature extracted method such as deep learning technique has been focus the studied for construction material images. Although DeGol *et al.*, [9] 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. In our proposed work, therefore, a new notable transfer learning scenario of CNN based method with its improvement is investigated for material image classification task. Where, two of pre-trained architectures that are GoogleNet and ResNet101 trained on ImageNet dataset are employed. Moreover, Autoencoder and PCA are also applied incorporated to the pre-trained features from GoogleNet and ResNet101.

III. CNN Based Methods

A. CNN Based Model

Emerging of CNN model comes from three of concepts, which are sparse interaction, parameter sharing, and

equivariant representation [10]. Such concepts transform to the network configuration demonstrates in Fig. 1. The network may view as it consists mainly of two processes that are feature learning process and classification process. In feature learning process, the principle stages, which are convolution, nonlinearity, and pooling stages incorporated to fully connected stage, are used in order extract the features in deep. Such stages are named respectively in Fig. 1 as *CONV*, *ReLU*, *POOL*, and *FC*. Where *ReLU* means the stage uses Rectified Linear Unit (ReLU) function as a nonlinearity function. In CNN model, many of these principle stages are consecutively arranged as layer-by-layer aimed at automatically learning the deep features from input. The features extracted from feature learning process will be further used for the classification process by Softmax function. It applied for the last layer of the network in order to give the output in probability manner.

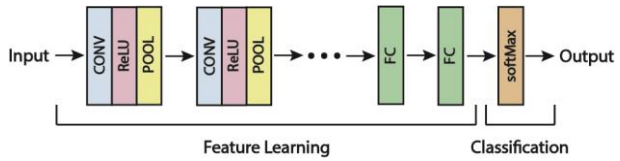


Figure 1. Principle layers of CNN model.

1) Convolution layer

In CNN model, convolution is a major operation. It is in the formed of 2D convolution with 3D input, where many filters of size $k \times k \times D$ are used once at a time to convolute with the 3D input of size $W \times H \times D$ by sliding windows manner. The convolution result from one filter is one of the feature map output. Therefore, when N filters are applied in a convolution layer, the entire output from the convolution stage is a stack of N feature maps. That means the information in each convolution layer of CNN model is viewed as the features in 3D.

2) ReLU layer

Because convolution is a linear operation, the feature maps result from the convolution layer always pass through a non-linear ReLU function in order to extract the non-linear property of the features. ReLU function is shown in (1). It simply but work very well by converting all the negative values of input to zero where keeps the others as the original.

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \gg 0 \end{cases} \quad (1)$$

3) Pooling layer

The pooling layer of CNN model used for subsampling to each feature map input. After the pooling stage, the dimension of width (W) and height (H) of the feature map will decrease. By subsampling, therefore, difference of pooling size can be used. If use the pooling size of 2×2, it means only one value from four values is selected as the subsampling value. Whereas, max pooling, average pooling, or any others pooling types can be employed to selected the one subsampling value from such four values.

4) Fully connected layer

Most CNN models use some of Fully Connected (FC) layers at the position near output. In FC layer, the feature maps from previous layer will be arranged as a vector data to be the input of the FC layer and connect together in the same way as a Multi-Layer Perceptron (MLP) network used.

5) Softmax layer

In feature learning process, many layers consisting of the convolution, nonlinearity by ReLU function, and pooling are consecutively arranged to form a network. Some other stages may include such as the stage of normalization or dropout that are also used in AlexNet.

For the classification process, the softMax function is employed to transform the output of the network to be the values in term of probability. Its function shows in (2).

$$softMax(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_j}} \quad (2)$$

where $softMax(y_i)$ is the softmax result of each y_i, y_i is an output of each neuron i, e^{y_i} is the exponential value of y_i, and k is the component of vector y.

When we want to apply CNN based method to our application task we can do in two different ways. The first way knows in term *learning from scratch*. By this scheme, the CNN structure that appropriates to the studied dataset (task specific dataset) is created and fully trains on such dataset. It may works very well if our task specific dataset is large enough. The second way is *transfer learning*. Where the knowledge (in term of weight and bias parameters) from some of the pre-trained architectures trained by other dataset that big enough (source task dataset) is transferred to the task specific dataset. 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-learning approach has been come up and many of well-known CNN pre-trained architectures are public and appreciation by the researchers in the fields.

B. CNN Pre-Trained Models

Nowadays, many so named CNN architectures pre-trained on some source tasks are public. Most of all, architectures involved each year ILSVRC are well known. Table I shows some of ILSVRC architectures that pre-trained by ImageNet dataset. A table focuses to compare their complexity in term of depth and total number of parameters. Where, number of parameters of CNN model means total number of weights and bias employed in a network.

TABLE I. COMPARE PRE-TRAINED ARCHITECTURES DETAIL THAT USED IMAGENET DATASET AS A SOURCE TASK

Architecture	No. Weighted Layers	No. Total Layers	No. of Parameters (million)
AlexNet	8	25	62.4
GoogleNet	22	144	6.8
ResNet50	50	177	25.6
ResNet101	101	347	44.5

For GoogleNet, Its weighted layers has around 3 times deeper than AlexNet [11] but total number of parameters has 10 times less than. Actually, after AlexNet, ZF Net [12] was also proposed by modifying some details of AlexNet. The number of layer of AlexNet and ZF Net are the same, therefore, we do not show for ZF Net in Table I. ResNet50 and ResNet101 were the two architectures based on ResNet structure. ResNet101 is deeper and as a result has more parameters. In this work, we select architectures of GoogleNet and ResNet101. These two architectures are differences both in deep and in detailed layers. They were per-trained by around 1.2 million images of 1000 classes of everyday used images. Details for each architecture explain as follow:

1) GoogleNet

Szegegy *et al.*, [1] from Google research term developed an architecture of GoogleNet 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*. Each module used only small filter size of 1×1, 3×3, and 5×5. Each block in a module can do in parallel and the results from all blocks are concatenated to be the inception module output send to the next layer in a network. GoogleNet used total number of parameters around ten times fewer than AlexNet as showed in Table I.

2) ResNet101

If compare by the layer in deep, ResNet101 is five times deeper than GoogleNet when count for its weighted layers. ResNet101 employed the same inside details as ResNet152 that won ILSVRC 2015. All ResNet architectures construct based on the core idea of introducing a so-called *identity shortcut connection* that skips one or more layers. Such an idea was then transformed to a *deep residual learning* framework [2].

C. Transfer Learning of CNN

Transfer learning of knowledge from CNN based models can apply by two schemes, which are *fixed feature extractor* and *fine-tuning*. Fixed feature extractor directly uses the pre-trained weights and bias and 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 using a task specific dataset with weights and bias initialized from transferring pre-trained weights and bias.

In practice, both fixed featured extractor and fine-tuning are popular for the image classification tasks. Due to CNN features are more generic in early layers and more original-dataset-specific in later layers, There are some common rules for navigating the following 4 scenarios [13]:

1) Task specific dataset is small and similar to source task dataset: employ fixed features extractor by training a linear classifier from the activation features at the top layer.

2) Task specific dataset is large and similar to source task dataset: employ fine-tuning through the full pre-trained network.

3) Task specific dataset is small but very different from the source task dataset: employ fixed features extractor by training a linear classifier from the activation features somewhere earlier in the network.

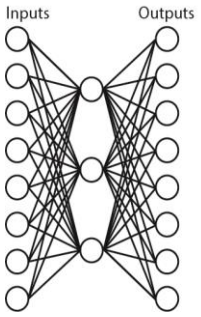
4) Task specific dataset is large and very different from the source task dataset: employ fine-tuning through the full pre-trained network. Actually, we can afford to train a CNN model from scratch.

IV. Feature Representation Methods

A. Autoencoder

Autoencoders are a type of neural network architecture that take in an input vector, compress (encode) the input to a reduced set of dimensions and then reconstruct (decode) the compressed data back to its original form. Therefore, a lossy transformation is applied to the data that may be used in applications like image compression [14]. Although concept of Autoencoders is as old as neural network, it comes up of interest since the emerging of deep learning. Autoencoders of many consecutive hidden layers is named as stack-Autoencoders for deep learning.

Example of the learning by Autoencoder shows in Fig. 2. This Autoencoder is forced to encode a 8 bits input to be 3 bits by setting an output of the network the same as an input. That mean, by Autoencoder it is forced to learn $f(x)=x$, where x is an input. Its weight values act as the encoding function (i.e. group of segments on the left of Fig. 2(a)) and decoding function are the weighted on the right side of Fig. 2(a). For the encoded values, just encoding weights are used. After activated such encoding weight to each input, we get the encoded 3 bits results by the values show as Hidden Values of Fig. 2(b).



(a) Autoencoder model

Input	Hidden Values	Output
10000000	.89 .04 .08	10000000
01000000	.15 .99 .99	01000000
00100000	.01 .97 .27	00100000
00010000	.99 .97 .71	00010000
00001000	.03 .05 .02	00001000
00000100	.01 .11 .88	00000100
00000010	.80 .01 .98	00000010
00000001	.60 .94 .01	00000001

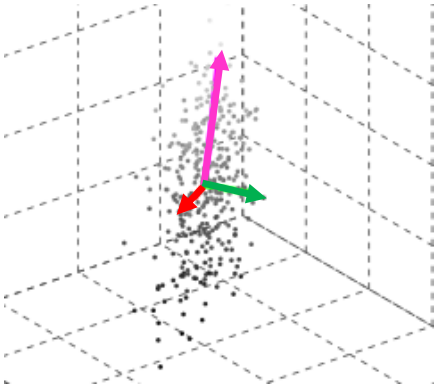
(b) Activated features learned at the hidden layer of (a).

Figure 2. Autoencoder model and its example [15].

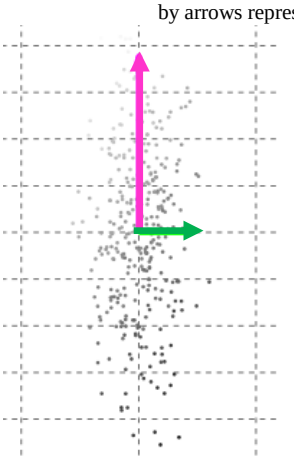
B. Principal Component Analysis (PCA)

PCA is a non-parametric method of extracting relevant information from data by reducing a complex data to a lower dimension. It is an unsupervised learning method for dimensionality-reduction.

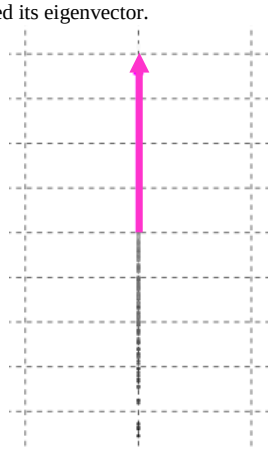
Example of PCA shows in Fig. 3. The original data with 3D feature space is in Fig. 3(a). Each principal component shows by arrows represented its eigenvector. After applied PCA to this data, if select the two largest components, will get the result as in Fig. 3(b). Whereas, if select only the one largest component, result will be as shown Fig. 3(c).



(a) Original data with 3D feature space. Each principal component shows by arrows represented its eigenvector.



(b) Project the original data in (a) onto the two largest eigenvectors and obtain a 2D feature space.



(c) Project the original data in (a) onto the largest eigenvector and obtain a 1D feature space.

Figure 3. Example of PCA. Edit from [16].

PCA process consists the five steps are as follows [17]:

1) Subtract the mean from each of the dimensions.
2) Calculate the covariance matrix.
3) Calculate the eigenvectors V and eigenvalues D of the covariance matrix.
4) Reduce dimensionality and form feature vector. Where, the eigenvector with the highest eigenvalue is the principal component of the data.
5) Derive the new data as FinalData. Where, FinalData = RowFeatureVector × RowFeatureMeanData

V. Proposed Works

Our proposed works can be divided into two parts. The first part, we explore on the transfer learning of GoogleNet and ResNet101 by both fixed feature extractor scheme and fine-tuning scheme. The second part, we employ Autoencoder and PCA as the feature representation methods to the transferred pre-trained feature results from the first part. Details of each part explain in Subsection A and Subsection B, respectively.

A. Transfer Learning of GoogleNet and ResNet101

1) Fixed feature extractor

By a scheme of fixed feature extractor, the weights and bias from the pre-trained architecture are directly transferred and used for the classification process by have no need to retrain the network with the training set of the task specific data set. Therefore, 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 task specific dataset is the construction material images that consists three classes of materials, which are brick, concrete, and wood. In order to evaluate the performance of fixed feature extractor by both GoogleNet and ResNet101, we use Support Vector Machine (SVM) as a classifier. We employ the feature from layer "pool5" for GoogleNet. For ResNet101 we use the last layer of feature learning process. The last layer before softMax layer.

2) Fine-Tuning

By fine-tuning scheme, we observed the fine-tuning parameters based on an empirical experiments follow a work of [18]. 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.

B. Feature Representation of the Transferred Pre-trained Features

For each CNN pre-trained architecture, the feature result from each transfer-learning scheme of subsection A is used as an input to Autoencoder and PCA. Actually, we should have four Autoencoder models and four PCA models. However, we experiment on only three Autoencoder models and three PCA models. We do not apply the feature representation method to the fine-tuning feature of ResNet101 because we believe the optimal fine-tuning result from the previous experiment still not reach due to the complex in setting the hyper-parameters in the fine-tuning process.

VI. Experimental Results and Discussions

A. Task Specific Dataset

Our studied dataset or task specific dataset is a collection of construction material images consists of three prominent classes of materials, which are brick, wood, and concrete. There are parts of the public images in a work of DeGol *et al.*, [9]. Examples of some images for each class show in Fig. 4. Class of brick is shown in Fig. 4(a), Concrete is Fig. 4(b), and Fig. 4(c) is wood, respectively. All images are 100×100 pixels resolution. The training set consists of 400 images per class and the testing set is 200 images per class.

(a) brick (b) concrete (c) wood

Figure 4. Examples of some images in each class.

B. Experimental Results

Table II shows percent of accuracy results from the transfer learning scenarios using GoogleNet and ResNet101 pre-trained architectures. The best accuracy from GoogleNet obtains when fine-tuning is applied; it is of 95.50%. By fine-tuning, the performance improves 3.33% from the fixed feature extractor scheme. In case of ResNet101, it exposes quite high in accuracy when just fixed feature extractor scheme is used; it is of 95.00%. On the other hand, by fine-tuning of ResNet101, it is difficult to tune this pre-trained model because of its depth. In total, it consists of 347 detail layers as shown in Table I. As a result, the fine-tuning process is complex in searching for the optimal hyper-parameters. Based on our experiences, the best result we get by fine-tuning is just 93.00% as shows in Table II. By the result, we expect it is not the optimal one.

TABLE II. ACCURACY RESULTS (%) FROM THE TRANSFER LEARNING OF GOOGLENET AND RESNET101 ARCHITECTURES

Pre-Trained Architecture	Transfer Learning Scheme		Improvement after fine-tuning
	Fixed Feature Extractor	Fine-Tuning	
GoogleNet	91.17	95.50	3.33
ResNet101	95.00	93.00 *	-2.00

Table III compares the accuracy results after the CNN based features from the transfer learning features of GoogleNet and ResNet101 are applied by PCA and Autoencoder. When the transferred fixed feature extractor of GoogleNet is employed as an input to PCA and Autoencoder, the classification results can improve to be 92.33 and 93.50, respectively. For ResNet101, PCA can improve the fixed feature extractor based feature to be 96.83%, whereas by Autoencoder it can improve up to 97.83% as shows by bold font value in Table III.

It is note that applying the feature representation methods to the transferred fine-tuning feature cannot significant improved the performance. Thus, show by the result of Table II when fine-tuning feature of GoogleNet is employed. By PCA, the result is less than when the based feature is used. Although by Autoencoder reveal a bit higher result, it is not the significant improvement. For fine-tuning feature of ResNet101, we do not conduct the experiment on feature representation methods because we believe the based fine-tuning features still not the optimal result.

TABLE III. COMPARE ACCURACY RESULTS (%) AFTER FEATURES FROM GOOGLENET AND RESNET101 ARE APPLIED BY PCA AND AUTOENCODER

Model/Representation Method	Based Features Used		Best Improvement from Based
	Fixed Feature Extractor	Fine Tuning	
GoogleNet (Based)	*(91.17)*	*(95.50)*	
PCA	92.33	93.67	
Autoencoder	93.50	95.83	2.33
ResNet101 (Based)	*(95.00)*	NA	
PCA	96.83	NA	
Autoencoder	**97.83**	NA	2.83

Fig. 5 represents the confusion matrix result from the testing set when Autoencoder applied to ResNet101 fixed feature extractor based features. The test set consists of 600 images from three classes. There are 200 image or 33.33% for each class. Target class labels in Fig. 5 is an actual class and output class is a class from the classification result of the proposed method. Overall accuracy is 97.83% (a matrix shows 97.80 by ceiling) marked by bold font. It is the highest classification result from our proposed work. When only per class classification is considered, it can classify concrete class with the highest accuracy of which 99.50% that label by italic bold font in Fig. 5. Out of 199 images from 200 images for a class of concrete can be correctly classified after Autoencoder is applied.

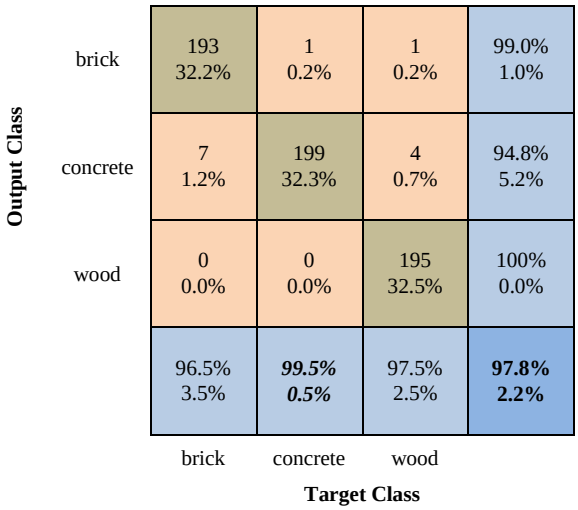


Figure 5. Confusion matrix result when Autoencoder applied to ResNet101 fixed feature extractor.

The confusion matrix shows in Fig. 6 is the result from the testing set when PCA is applied to ResNet101 of which fixed feature extractor is used as a based feature. Overall accuracy is 96.83% (a matrix shows 96.8 by ceiling) marked as bold font. For per class classification, class of wood can classify with the highest accuracy of which 99.50% represent as italic bold font.

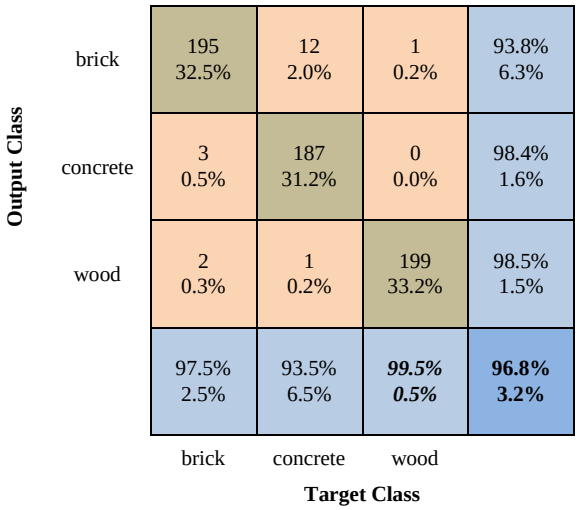


Figure 6. Confusion matrix result when PCA applied to ResNet101 fixed feature extractor.

C. Discussions

Our task specific dataset is a small one. It consists of only 1,200 training images and 600 testing images. According to the common rules for navigating the 4 scenarios of transfer learning in Subsection C of Section III (CNN Based Method), just applying fixed feature extraction scheme, it should get the good result. From our experiment, according the results show in Table II. When used fixed feature extractor of ResNet101, the classification result is better than fixed feature extractor of GoogleNet. Where, GoogleNet can improve its performance by the fine-tuning scheme. In a case of ResNet101, we believe we still not reach the optimal fine-tuning result due to the complex in setting the hyper-parameters in the fine-tuning process.

By the way, the performance from fixed feature extractor of both pre-trained model can be further improved when apply Autoencoder and PCA to such feature. By Autoencoder based representation method reveals the performance can improve more than PCA in all cases as shown in Table III. The best improvement of GoogleNet is of 2.33%, whereas for ResNet101 it is of 2.83%. Best of all, Autoencoder when use ResNet101 fixed feature extractor as a based feature get the highest performance for our task specific dataset of construction material image classification. It can also be inferred, just applying Autoencoder on top of the pre-trained transferred features, the performance can be improved by we have no need to fine-tune the complex pre-trained model.

From the classification results of per class classification shown by the confusion matrices; Fig. 5 and Fig. 6, we can see that Autoencoder applied to ResNet101 fixed feature extractor can classify very well for a class of concrete. Whereas, when PCA applied to ResNet101 fixed feature extractor it is good for classifying the class of wood. For our further research, we therefore, may investigate on combining of the two cases of features.

VII. Conclusion

In our study, the state of the art approach based on the transfer learning of CNN pre-trained architectures is investigated. A set of construction material images is act as a task specific dataset in the transfer-learning scenario. We investigate on the transfer learning of GoogleNet and ResNet101 that pre-trained on ImageNet dataset (source task). By applying both of the transfer-learning schemes, they reveal quite satisfied results. The best for GoogleNet, it gets 95.50 percent of the classification accuracy by fine-tuning scheme. Where, for ResNet101, the best is of 95.00 percent by using fixed feature extractor scheme. Nevertheless, after the learning based representation methods are further employed on top of the transferred features, they expose more appeal results. By Autoencoder based representation method reveals the performance can improve more than PCA (Principal Component Analysis) in all cases. Especially, when the fixed feature extractor of ResNet101 are used as the input to Autoencoder, the classified result can be improved up to 97.83%. It can be inferred, just applying Autoencoder on top of the pre-trained transferred features, the performance can be improved by we have no need to fine-tune the complex pre-trained model.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The first author conducted the research, organized the research framework, experimented and prepared the manuscript. The second author validated the research framework and edited the manuscript. The last author recommended for the experimentation steps and validated the experimental results.

References

[1] C. Szegedy, *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, 2014.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Computer Vision and Pattern Recognition*, 2015.

[3] K. Chen, W. Lu, Y. Peng, S. Rowlinson, and G. Q. Huang, "Bridging BIM and building: From a literature review to an integrated conceptual framework," *International Journal of Project Management*, vol. 33, pp. 1405-1416, 2015.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] J. DeGol, M. Golparvar-Fard, and D. Hoiem, "Geometry-Informed material recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1554-1562.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1106-1114.

[12] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2013, pp. 818-833.

[13] Transfer learning. [Online]. Available: http://cs231n.github.io/transfer-learning/

[14] T. M. Dahan, "PCA and Autoencoders," Technical Report, Concordia University, INSE 6220 - Fall 2017.

[15] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997, ch. 4, pp. 106-108.

[16] V. Spruyt. Feature extraction using PCA. [Online]. Available: http://www.visiondummy.com/2014/05/feature-extraction-using-pca/

[17] S. Y. Elhabian and A. Farag, "A tutorial on data reduction: Principal component analysis theoretical discussion," Technical Report, Computer Vision and Image Processing Laboratory, CVIP Lab, University of Louisville, September 2009.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural network?" in *Proc. NIP*, 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, 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, master degree in Computer Science from the Prince of Songkla University, Thailand in 1991 and 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, master degree in Computer Science from the Prince of Songkla University, Thailand in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A. in 1999. His current research includes Data mining, Artificial Intelligence, Computational Statistics.