ISSN 2288-4866 (Print)
ISSN 2288-4882 (Online)
http://www.jiisonline.org
J Intell Inform Syst 2018 March: 24(1): 205~225 http://dx.doi.org/10.13088/jiis.2018.24.1.205

Transfer Learning using Multiple ConvNet Layers Activation Features with Principal Component Analysis for Image Classification*

Batkhuu Byambajav
Department of Computer Engineering, Inha University
(*bybatkhuu@inha.edu*)

Jumabek Alikhanov
Department of Computer Engineering, Inha University
(*jumabek4044@gmail.com*)

Yang Fang
Department of Computer Engineering, Inha University
(*fangyang968@gmail.com*)

Seunghyun Ko
Department of Computer Engineering, Inha University
(*kosehy@gmail.com*)

Geun Sik Jo
Department of Computer Engineering, Inha University
(*gsjo@inha.ac.kr*)

Convolutional Neural Network (ConvNet) is one class of the powerful Deep Neural Network that can analyze and learn hierarchies of visual features. Originally, first neural network (Neocognitron) was introduced in the 80s. At that time, the neural network was not broadly used in both industry and academic field by cause of large-scale dataset shortage and low computational power. However, after a few decades later in 2012, Krizhevsky made a breakthrough on ILSVRC-12 visual recognition competition using Convolutional Neural Network. That breakthrough revived people interest in the neural network. The success of Convolutional Neural Network is achieved with two main factors. First of them is the emergence of advanced hardware (GPUs) for sufficient parallel computation. Second is the availability of large-scale datasets such as ImageNet (ILSVRC) dataset for training. Unfortunately, many new domains are bottlenecked by these factors. For most domains, it is difficult and requires lots of effort to gather large-scale dataset to train a ConvNet. Moreover, even if we have a large-scale dataset, training ConvNet from scratch is required expensive resource and time-consuming. These two obstacles can be solved by using transfer learning. Transfer learning is a method for transferring the knowledge from a source domain to new domain. There are two major Transfer learning cases. First one is ConvNet as fixed feature extractor, and the second one is Fine-tune the ConvNet on a new dataset. In the first case, using pre-trained ConvNet (such as on ImageNet) to compute feed-forward activations of the image into the ConvNet and extract activation features from specific layers. In the second case, replacing and retraining the ConvNet classifier on the new dataset, then fine-tune the weights of the pre-trained network with the backpropagation. In this paper, we focus on using multiple ConvNet layers as a fixed feature extractor only. However, applying features with high dimensional complexity that is directly extracted from multiple ConvNet layers is still a challenging problem. We observe that features extracted from multiple ConvNet layers address the different characteristics of the image which means better representation could be obtained by finding the optimal combination of multiple ConvNet layers. Based on that observation, we propose to employ multiple ConvNet layer representations for transfer learning instead of a single ConvNet layer representation. Overall, our primary pipeline has three steps. Firstly, images from target task are given as input to

* This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2015-R1A2A2A03006190).

ConvNet, then that image will be feed-forwarded into pre-trained AlexNet, and the activation features from three fully connected convolutional layers are extracted. Secondly, activation features of three ConvNet layers are concatenated to obtain multiple ConvNet layers representation because it will gain more information about an image. When three fully connected layer features concatenated, the occurring image representation would have 9192 (4096+4096+1000) dimension features. However, features extracted from multiple ConvNet layers are redundant and noisy since they are extracted from the same ConvNet. Thus, a third step, we will use Principal Component Analysis (PCA) to select salient features before the training phase. When salient features are obtained, the classifier can classify image more accurately, and the performance of transfer learning can be improved. To evaluate proposed method, experiments are conducted in three standard datasets (Caltech-256, VOC07, and SUN397) to compare multiple ConvNet layer representations against single ConvNet layer representation by using PCA for feature selection and dimension reduction. Our experiments demonstrated the importance of feature selection for multiple ConvNet layer representation. Moreover, our proposed approach achieved 75.6% accuracy compared to 73.9% accuracy achieved by FC7 layer on the Caltech-256 dataset, 73.1% accuracy compared to 69.2% accuracy achieved by FC8 layer on the VOC07 dataset, 52.2% accuracy compared to 48.7% accuracy achieved by FC7 layer on the SUN397 dataset. We also showed that our proposed approach achieved superior performance, 2.8%, 2.1% and 3.1% accuracy improvement on Caltech-256, VOC07, and SUN397 dataset respectively compare to existing work.

Key Words : Deep Learning, Transfer Learning, Fixed Feature Extractor, Feature Selection, Image Classification

Received : December 5, 2017 Revised : February 5, 2018 Accepted : February 27, 2018
Publication Type : Regular Paper Corresponding Author : Geun Sik Jo

1. Introduction

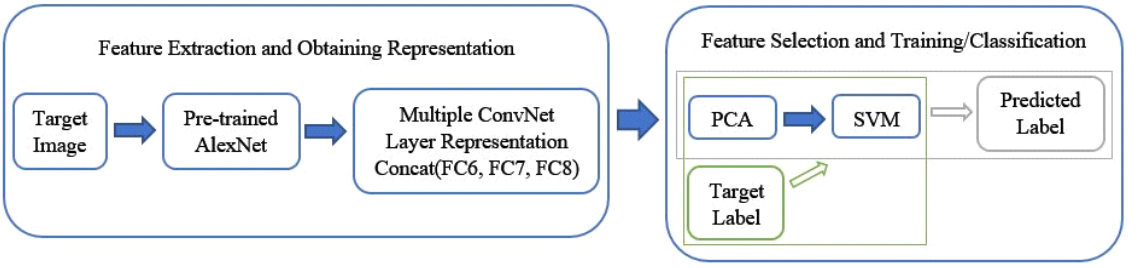
Convolutional Neural Networks (ConvNets) are potent models that learn hierarchal features of visual data, that could also be used to obtain image representation for transfer learning. Originally, Neocognitron, the ancestor of neural network (Fukishima, 1990) was introduced back in the 80s. Back then, the neural network was not widely used in both academia and industry field due to lack of large-scale dataset and computational power. However, in 2012, Krizhevsky (Krizhevsky et al., 2012) made a breakthrough on ILSVRC-12 (Russakovsky et al., 2015) visual recognition challenge. That breakthrough rekindled people interest in the neural network. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset and advancement of parallel computing hardware (GPUs) were the main factors of Krizhevsky's win. Thus, we can conclude that recent success of Deep Convolutional Neural Networks is achieved through two essential factors. Firstly, the emergence of GPUs for parallel computation. Secondly, the availability of large-scale datasets such as ImageNet. Unfortunately, these factors can become a bottleneck for lots of domains. For most domains, gathering large-scale dataset to train the ConvNet requires high effort. Moreover, even if we have a large-scale dataset, training ConvNet from scratch is required expensive resource and time-consuming. These two obstacles can be

solved by using transfer learning. Transfer learning transfers the knowledge learned from the source domain to target domain. The primary pipeline of transfer learning is to first train the ConvNet in a source domain (source task) where large-scale dataset exists. The second step is to use pre-trained ConvNet as a feature extractor for the new domain (target task) where the dataset is small.

Previous works (Donahue, 2013), (Girshick et al., 2014), (Oquab et al., 2014), (Razavian et al., 2014), (Zeiler et al., 2014), (Azizpour et al., 2014) showed transfer learning by using specific layer activation features of ConvNet as an image (signal) representation for the target task. They extract ConvNet features from the particular layer, for instance, fully connected layer 6 (FC6) of AlexNet and train linear SVM classifier using features obtained from FC6. Existing works on transfer learning apply only specific single layer activation features of the ConvNet as image representation. However, ConvNets learn features in hierarchically manner. Thus, compared to layer FC6, layer FC7 will generate higher level activation features and address different aspects of images that FC6 cannot address.

Recently, transfer learning methods widely used for prediction and classification fields. Therefore, transfer learning can have applied for business applications such as stock exchange prediction (Lee et al, 2017), study of investment model (Song et al, 2017), classifying targeted customer for advertising, or financial fraud detection (Sukjae et al, 2017). In the financial sector, stock trading systems are improved with machine learning methods among artificial intelligence fields. For instance, in stock exchange market, if stock trading system already learned about fruit companies' past stock market information, we apply that learned model to predict specific apple's future stock price with less training time. In marketing and advertising sector, if we have advertising recommender system which uses machine learning model that good at predicting car interested customers, we can use that model to predict Ford car interested customers.

In this paper, we first present the idea of the diversity of activation features for encoding on different aspects of the image signal. In other words, different ConvNet layer features to address the different characteristic of an image to some level. This notion is demonstrated and explained in Chapter 3. Once we have that notion, we will study using multiple ConvNet layer features rather than single ConvNet layer features will produce better outcomes. Unlike other works, we are proposing to apply multiple ConvNet layer representations instead of a single ConvNet layer representation. However, combining those multiple ConvNet layer features will cause our feature space more complex than simple individual ConvNet layer features. Furthermore, redundant and noisy features from each layer will be combined and that leads to decrease in performance. Hence, we need feature selection to reduce the feature space complexity and get rid of noisy features. We select features implicitly by using PCA. The primary workflow process is illustrated in <Figure 1>. Our system (1) takes the



〈Figure 1〉 Our method workflow for Transfer Learning on Multiple ConvNet Layers

input of target task images, (2) extracts activation features from fully connected layers 6, 7, and 8 of AlexNet to obtain three individual image representations of FC6, FC7, and FC8 respectively, (3) concatenates features of those three layers to gain multiple ConvNet layer representation (FC6-FC7-FC8), (4) perform PCA as feature selection and train SVM classifier on both multiple and individual ConvNet layer representations. Our contribution is two-fold. First, show the superior performance of multiple layer representation over single layer representations. Second, we use PCA (Principal Component Analysis) to select only robust and distinct features that are beneficial towards classification from concatenated multiple ConvNet layer features.

The rest of the paper is designed as follows. Chapter 2 discusses the background and related works. Main theory, methods, and algorithms are described in Chapter 3. The experimental results will be examined in Chapter 4. Finally, Chapter 5 covers conclusions and future works.

2. Background and Related Works

2.1 Convolutional Neural Networks Architecture

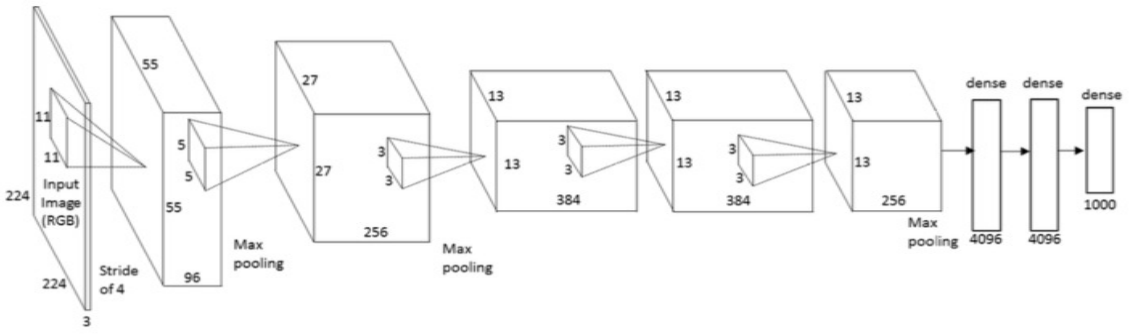
The Neural Network receives an input and transforms it into a series of hidden layers. Every hidden layer is formed up of some set of neurons, every neuron is fully connected to whole neurons in the prior layer, and neurons in the single layer function utterly independent without sharing any connections. The last fully connected layer is named “output layer,” and it depicts the class scores. The problem with conventional neural networks is that they do not scale appropriately to the full-size image. For example, in the CIFAR-10 dataset (Krizhevsky et al., 2009) images are in very small size 32x32x3 (height, width, and color channels). That means individual fully connected neuron in primary hidden layer would have 32*32*3=3072 parameters (weights). We have multiple of such neurons in that first fully connected layer. Furthermore, we have multiple of such layers. Though, that amount of weights could be learned for that size of images. Despite, when

image size raises, state 224x224 training conventional neural network will result in overfitting because of a significant amount of weights ratio over the whole of training samples. The consequence message is that this full connectivity is wasteful and the vast number of weights (parameters) would immediately drive to overfitting. Hence, we use the Convolutional Neural Network to evade full connectivity by weight sharing technique.

Convolutional Neural Networks or ConvNet consists of neurons that have learnable biases and weights. Each neuron will accept some inputs, performs a dot product and optionally followed by non-linearity. The whole network represents a single differentiable score function: from input image pixels on individual point to class scores at another. ConvNet has loss function at the end of the fully-connected layers which back-propagates the error while the training process. Convolutional Neural Network takes benefit of the fact that input dwells of images and they restrain the architecture more sensibly. In particular, unlike conventional neural networks, the layers of ConvNet have neurons arranged in three dimensions: height, width, and depth. For instance, the input images of the CIFAR-10 dataset are an input capacity of activations, and the volume has dimensions 32x32x3 (height, width, and depth respectively). The neurons in the layer will only be connected to the small area of the layer before it, rather than all neurons in a fully connected way. Furthermore, the final layer would have 1x1x10 dimensions for the CIFAR-10 dataset, because the end of the ConvNet architecture will reduce the entire image into a single vector of class scores, ordered along the depth dimension.

2.1.1 AlexNet Architecture

With the advent of large-scale datasets and GPU hardware that will enable massive parallel computation power, Krizhevsky (Krizhevsky et al., 2012) won the ILSVRC-12 (Russakovsky et al., 2015) image classification competition with a large margin. Unlike other participants, Krizhevsky employed Neural Network, precisely Convolutional Neural Network. As depicted in <Figure 2>, the AlexNet contains eight layers; first five are convolutional layers and remaining three are fully-connected layers. The output of the latest fully connected layer is supplied to softmax which generates distribution over the 1000 class labels. AlexNet increases the multinomial logistic regression objective to maximum, which is similar to maximizing the mean covering training cases of log-probability of the correct label following the prediction arrangement. The first convolutional layer filters 224x224x3 input image with a stride of 4 pixels with 96 kernels of size 11x11x3. Stride is the distance between the receptive field of two neighboring neurons in kernel map. The second convolutional layer filters with the 5x5x96 size of 256 kernels. The third convolutional layer filters with 384 kernels of size 3x3x256. The fourth convolutional layer filters with 384 kernels of size 3x3x384 and the fifth convolutional layer filters with 256 kernels of size 3x3x384. Following two



〈Figure 2〉 Illustration of AlexNet CNN

fully connected layers, 6 and 7 have 4096 neurons. The final fully connected layer has 1000 neurons. Note, original architecture in (Krizhevsky et al., 2012) was designed for two GPUs since this architecture is designed for one GPU for some layers size of the kernels might be the twice large as in original architecture.

2.2 Transfer Learning

In reality, very few people train entire Convolutional Network from scratch, because it is approximately rare to have a dataset of sufficient size. Instead, it is typical to pre-train ConvNet on a large dataset (such as ImageNet), and then apply the ConvNet either as an initialization or fixed feature extractor for the target task. There are two major Transfer Learning scenarios:

ConvNet as fixed feature extractor: Take pre-trained ConvNet, compute feed-forward activations of the image into the ConvNet and extract activation features from specific layers. It is observed that last few fully connected layers of AlexNet conduce to serve as a useful feature extractor for classification. This way, ConvNet is treated as fixed feature extractor for new dataset. In the AlexNet, to extract activation features from FC8, would compute feed forward computation of AlexNet and receive activation features from the FC8 layer which is a 1000-D vector for every image. Once when we extracted the 1000-D features for all images, we need to train a linear classifier for the new dataset.

Fine-tune the ConvNet: The second strategy is not only by replacing and retraining the ConvNet classifier on the new dataset but also by fine-tuning the weights of the pre-trained network with the backpropagation. It is feasible to fine-tune whole layers of the ConvNet, or it is feasible to keep some of the initial layers fixed and only fine-tune some higher-level of the network. That is motivated by the observation that the preceding features of a ConvNet contain more generic features that should be helpful to many other tasks, but end layers of the ConvNet become continuously more specific to the details of the

classes.

2.3 Principle Component Analysis

In original paper of Principle Component Analysis (Abdi et al., 2010), they mentioned that it would be used to emphasize variation and bring out great patterns in a dataset (features). In generally, PCA applies a vector space converts to reduce the dimensionality of large data sets. It is often useful to measure data regarding its principal components rather than on a standard x-y axis. Principal components are the underlying structure of the data. They are the directions where most variance, which means data is most spread out in that directions. Imagine that ConvNet layer features are points of data. To obtain most variance directions, find the straight line that the data is most spread out onto it. Utilizing mathematical projection, the original data set, which have many variables, can be interpreted in just a few variables. We can use math to find the principal component by eigenvectors and eigenvalues. While we obtain a set of data points, like the ConvNet layer features, we can interpret the set into eigenvectors and eigenvalues. Eigenvectors and eigenvalues exist in pairs: every eigenvalue has a corresponding eigenvector. An eigenvector is a direction; an eigenvalue is a number that tells us how much variance in that data and how to spread out the data is. The eigenvector with the highest eigenvalue is, hence the principal component. Number of eigenvectors and eigenvalues that exist equals the number of dimensions that dataset has. In order to get eigenvectors and eigenvalues, we need to calculate covariance matrix. Consider a data matrix X (in our case ConvNet layer features), each of n rows expresses different repetition of an experiment, and each of p columns gives a kind of feature. X^TX itself can be perceived as proportional to empirical sample covariance matrix of dataset X, and the eigen decomposition is derived from the covariance matrix. After eigen decomposition, we need to select principal components which are to maximize variance. (First component):

$$w_{(1)} = argmax\,\{\frac{w^T X^T X w}{w^T w}\} \tag{1}$$

Then compute further components (2). The *k*th component from *X:*

$$\hat{X}_k = X - \sum_{s=1}^{k-1} X w_{(s)} \mathrm{w}_{(s)}^{\mathrm{T}} \tag{2}$$

Finally, we can compute new features by projection matrix from components. In our case we need eigenvectors of extracted features from training datasets.

2.4 Related Works

In (Zeiler et al., 2014) authors improved the ConvNet architecture of Krizhevsky (AlexNet) and analyzed it with their visualization technique. They further showed how their ConvNet trained on ImageNet. They trained SVM on ConvNet layer and demonstrated that activation features extracted

from the end layers (e.g., FC6, FC7) of ConvNet provide a robust performance for the target task. Following the same pipeline, other research (Donahue et al., 2013), (Oquab et al., 2014), (Razavian et al., 2014), (Azizpour et al., 2014) performed transfer learning by training a linear SVM on certain layer features of ConvNet. In (Razavian et al., 2014) authors conducted experiments on a series of visual recognition tasks using CNN codes that are extracted from FC6 of AlexNet as image description. The experiments consistently produced superior results, compared to the state-of-the-art, extremely tuned approaches that do conditional handcrafted features alike HOG, SIFT, and LBP. They also showed that simple augmentation techniques, boost performance significantly. In all these works, they used SVM with single ConvNet layer features. Researchers showed that for most of the classification in transfer learning representation of FC7 of AlexNet will provide the reliable performance. Hence, they trained SVM on FC7 ConvNet features to employ in transfer learning for their tasks. Despite, in our paper, we aim to demonstrate the superior performance of combined ConvNet layers against individual ConvNet layer. The objective of combining multiple ConvNet layer features is to improve encoded signal knowledge. Put it in short; we concatenate multiple ConvNet layer features to obtain better image representations compared to single ConvNet layer representation. After we concatenate multiple ConvNet layer features, following feature space becomes a more complex cause of higher dimension compared to individual ConvNet layer. Because we concatenated multiple ConvNet layer features of the same ConvNet features, it makes obtained representation to redundant and noise. That makes some form of feature selection necessary. We use PCA with dimension reduction to tackle the problem of feature selection before training. As observed in previous works, the activation features from the last layers of ConvNet will provide reliable performance in transfer learning.

3. Multiple ConvNet Layers Activation Features for Transfer Learning as a Fixed Feature Extractor

In this chapter, we will show some observation to explain the difference between two ConvNet layers when it comes to encoding knowledge about the image data. We refer ConvNet layer features to activation features of a ConvNet layer. In the case of AlexNet, we claim that FC6 layer features and FC7 layer features are different from each other as they encode the different characteristics of an image. That means FC6 addresses the different aspect of the image that FC7 does not encode and vice versa. Although FC6 features overall function better compared to other layers in one specific task, some other characteristics are best encoded in the FC7 layer rather than the FC6 layer. We propose that obtaining image representation from all fully connected layers gives better performance compared to individual layer representation such as

FC6.

In section 3.1 we will provide our motivation for using combined multiple ConvNet layers features as image representation. In section 3.2 we present some technical challenges, when using combined multiple ConvNet layer representation and we will contribute a solution for this challenge.

3.1 Different Characteristics of Multiple ConvNet Layers

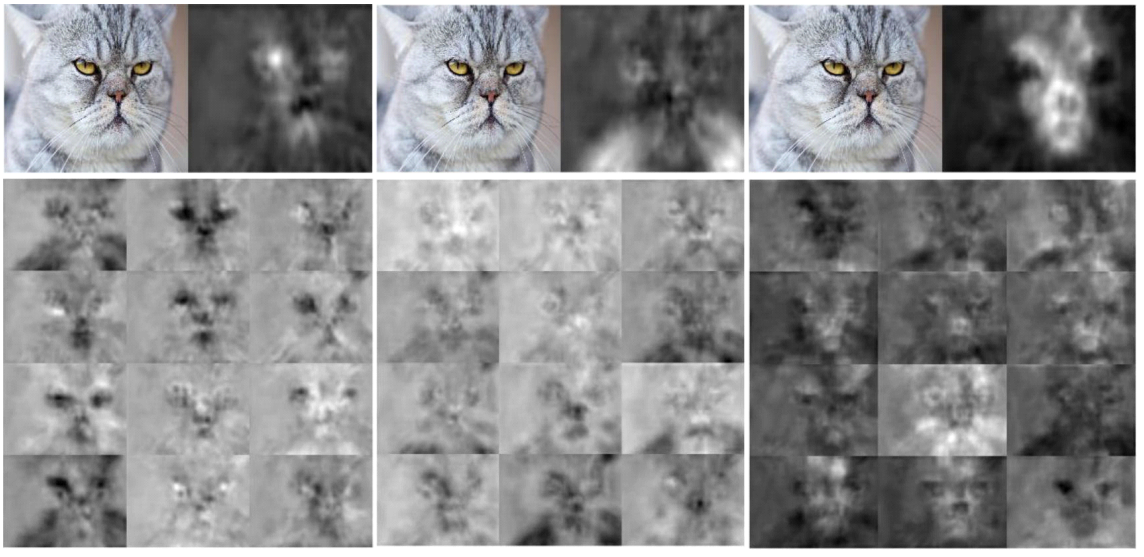
ConvNets hierarchically learn features. That is done by building more complex, abstract features as it goes from lower to higher layers. Several works on transfer learning have shown that specific fully connected layer features, in particular, FC7, gives the best performance when used for classification tasks. Hence, researchers favor using FC7 as image representation for their target task. Despite, if we recognize that ConvNet layer features are hierarchical, then FC8 layer defines some vital aspect of an image which FC7 does not. For, we wanted to examine how much different ConvNet layer features in the description in term of encoding image characteristics. To make our goal clear, consider the following instance. Suppose we have dataset *Ds*, where its test set *Ds-test* has five images (*Ds-test={"dog.jpg," "cat.jpg," "sheep.jpg," "camel.jpg" "horse.jpg,"}).* Names of the test examples resemble as their ground truth labels. For example, the label of dog.jpg is *"dog."* Applying the training set *Ds-train* of dataset *Ds*, train two classifiers C1, C2 employing FC7, FC8 representations respectively. Then, consider in testing phase C2 predicts test images *"dog.jpg," "cat.jpg,"* and *"sheep.jpg"* correctly out of five images whereas C1 predicts *"dog.jpg," "sheep.jpg"* and *"horse.jpg"* correctly. From the predictions outcomes, we can conclude that FC7 features are particularly useful for defining an image of cats whereas FC8 images are particularly useful for defining images of horses. This characteristic suggests the feasibility of obtaining better image representation by combining FC7 with FC8 features which are beneficial at defining images four classes instead of three. If somehow, we manage to do it, we could achieve four (*"dog.jpg," "cat.jpg," "sheep.jpg"* including *"horse.jpg,"*) accurate predictions for *Ds-test*. Thus, instead of lower (3/5*100) accuracy, now our classifier would gain higher (4/5*100) accuracy.

Our research is to investigate if above hypothetical instance indeed right for transfer learning on real-world datasets. If it is correct, then we will answer the question of "How to benefit from multiple ConvNet layers features and how to solve rising challenges?". First let us repeat the above example for real-world dataset Caltech-265 (Griffin et al., 2017). We split the dataset to training and test set as was done in (Griffin et al., 2017). Moreover, we will demonstrate each ConvNet's fully-connected layers activation features to explore difference of ConvNet FC layers.

To better understand what kind of information the feature maps contains in each fully-connected layer, we use Google's DeepDreamImage function (Szegedy et al., 2015) to visualize feature maps from layer FC6, FC7, and FC8, respectively.

As we can see above (<Figure 3>), layer FC6 features contain part shape information of different objects, such as the head of the dog, part body shape of birds, layer FC7 features not only can detect the part information but also contains more rich color information, and layer FC8 features contain the category information and holistic information of 1000 object classes, which are very powerful at distinguishing objects. Moreover, we also analyze the activation results of FC6, FC7, and FC8 given an input image of cat <Figure 4>.

〈Figure 3〉 Visualization of feature maps from layer FC6, FC7, and FC8



〈Figure 4〉 Visualization of activation features from layer FC6, FC7, and FC8

From left to right are activation results of layer FC6, FC7, and FC8. Moreover, from top to bottom are the maximum activation of each layer, activation results of 12 channel feature map randomly chosen among all channels, and all activation results for layer FC6, FC7, and FC8, respectively. As is shown at first row above, when we feed a cat image into the AlexNet, the maximum activation of layer FC6 among 4096 channel of feature maps strongly activates the left eye position, it means that features of the channel have learned information of cat's left eye. Moreover, maximum activation from layer FC7 highly activates the information of the lower parts of its face but not eyes and FC8 layer are powerful activates the holistic face positions, it demonstrates that layer FC8 have learned the entire features of the object, not only part information. The second rows show some activations from 12 channels of feature maps; it shows that layer of FC6 and FC7 are good at learning part based features whereas layer FC8 is good at learning the holistic object features. Moreover, last row shows all activations of each fully-connected layer. So intuitively, we can draw the expectation that if we combine the feature maps from multiple fully connected layers, the classifiers learned from multi-layer features should predict much better classification results than each of them independently. Note that not all the features in ConvNet layer are helpful towards to defining the image. Somewhat it is trained on source task, and now we are operating transfer learning by using pre-trained ConvNet as fixed feature extractor. For this reason, if we encode the image with FC8 representation which has 1000 dimensions, there will be some noise features. Therefore, if multiple ConvNet layers features combined by merely concatenating them, resulting in representation might become even worse than single layer representation because of the increase in some useless features and emergence of new repetitive features. Hence, we need feature selection to benefit from multiple ConvNet layers features. We solve the problem of feature selection using PCA where its eigenvectors with eigenvalues will select only useful features.

3.2 Principal Component Analysis for Feature Selection

While the behavior of ConvNet layers differs, it does not mean all of the 1000-dimensional features of FC8 encode distinct characteristics of an image, concerning the other 4096-dimensional features of FC6 and also 4096-dimensional features of FC7. Instead, a small number of features from FC6, FC7, and FC8 together would form a complementary feature. For instance, let's say only 3000 features out of 4096 are useful and distinct from each layer (let's say in FC8 case, 800 features are useful out of 1000). That means when we concatenate those two FC6, FC7 and FC8 ConvNets layers to get the final 6800-dimensional feature description, we will get a moderately better image description compared to applying individual ConvNet layer features. However, most of the features will be redundant, and not helpful towards defining an image. Some features might even

become a noise. Considering not all of the combined features are useful for defining the image, we encounter the problem of selecting only distinct and helpful features. In other words, we need to discover some method that will select only essential features from among 9192-dimensional features that are helpful for classification. For to select only helpful features, we use Principal Component Analysis as a feature selection. PCA selects a stable pattern of features that will decrease the loss. In other words, before training with SVM, PCA will select the best features, and that will give higher classification accuracy. This way, PCA will select only those distinct and helpful features from amongst the 9192-dimensional features. Hence, we use PCA to take advantage of multiple ConvNet layer features. PCA and eigenvectors also help dimension reduction. The PCA could be employed to reduce the dimensions of a dataset. It reduces the data down to its essential components, stripping away any unnecessary parts. According to the rule of thumb, we reduced 85% of dimensions to fasten training performance. More specifically, we reduced feature dimension to 7813 by PCA, which is one of the local optimal. We also tried few more feature reductions to find the optimal one. For example, when we reduced feature to 50% (4596), it gives 36.5% accuracy on the Caltech-256 dataset with multiple ConvNet layers representation, which is a worse result than 85% feature reduction. Given the fact that, decreasing too much feature dimensions will cause lack of important features, which is essential to classify. Along with another percentage of reductions (35%, 50%, 60%, and 75%), these reductions do obtain lower accuracy. However, when performing PCA without any dimension reduction, gives 70.6% accuracy on the Caltech-256 dataset with multiple ConvNet layers representation, which is close to 85% reduction, but not optimal. In future works, we can find global optimal dimension reduction by the k-nearest neighbor algorithm.

In summary, by using PCA with SVM, best features that are helpful for classification will be chosen. As we will survey in Chapter 4 this feature selection of PCA will result in the superior performance of multiple ConvNet layers representation against single ConvNet layer representation.

3.3 Implementation

In this paper, we used Matlab for visualization of activation features and also used open source Caffe (Jia et al., 2014) and publicly available AlexNet (Krizhevsky et al., 2012) pre-trained model from Caffe's Model Zoo to extract ConvNet layer activation features. Single layer representations extracted from FC6, FC7, and FC8 of AlexNet respectively. For multiple ConvNet layer representation, we merely concatenate three fully connected layer activation features. Note, PCA makes feature selection before the training. Therefore, here we merely concatenate multiple ConvNet layer representations without bothering about feature selection. Features are collected by averaging ConvNet-layer activations of 12 samples

of the original image. We refer readers to (Jia et al., 2014), (Krizhevsky et al., 2012) for more detailed information.

3.4 Runtime

We evaluated runtime performance on Intel Xeon 3.30GHz x 8 CPU, NVIDIA Tesla K40 GPU, 20GB RAM, and Ubuntu 14.04 64-bit. To training process, first we need to extract features from the pre-trained AlexNet model, it takes about approximately 15 minutes on the Caltech-256 dataset (For training set, 45GB), which is depends on dataset size. After that, we need to perform PCA as feature selector, it takes about 10~30 minutes, which is dependent on extracted feature size and dimension reduction number. As we mentioned in section 3.2, our classification accuracy depends on PCA dimension reduction number. Finally, we need to train SVM classifier, that takes around 1~2 days.

4. Experimental Results

Experiments were conducted to evaluate and to demonstrate the performance of the proposed method dealing with noisy and redundant features from multiple ConvNet layers. We compare the performance of multiple ConvNet layer representation against single ConvNet layer representation. As explained in Chapter 3, for feature selection, PCA is applied.

4.1 Datasets

We conducted our experiments on three standard classification datasets.

Caltech-256: Caltech-256 (Griffin et al., 2017) contains around 30,000 images with 257 categories, including a cluttered category. Each category contains at least 100 images. Following the lead of (Griffin et al., 2017), we split the dataset by taking random 60 images from each class for training set and the rest for the test set.

SUN397: SUN397 (Xiao et al., 2014) is one of the challenging datasets for scene classification. It contains 108,000 images of 397 categories. Each category contains at least 100 images. Following (Xiao et al., 2014) we divide the dataset to training and test set by ranking random 50 images for both training and test datasets.

VOC07: VOC07 (Everingham et al., 2015) training and validation sets contain 5011 images in total, and test set contains 4952 images. We used training and validation sets for our training phase (Razavian et al., 2014), (Azizpour et al., 2014).

4.2 Analysis of Experimental Results

First, we will explain how we conducted our experiment; then we will analyze and discuss the results. As explained before in Chapter 2, about transfer learning, there are two main cases. The first case is to apply pre-trained ConvNet as fixed feature extractor. In the second case, pre-trained ConvNet is fine-tuned on the new dataset. In this paper, we only concentrate on fixed feature extractor of transfer learning. Our purpose of this

chapter is to show the superior performance of multiple ConvNet layer representation to facing single ConvNet layer representation and present the analysis of the results.

We first experiment by using linear SVM without PCA. As we can observe from <Table 1>, for the Caltech-256 dataset, FC7 layer representation give a better result than FC6-FC7-FC8 layer representation. While for the SUN397 dataset, FC6 layer representation gives the better result than FC6-FC7-FC8 representation. Although for VOC07 dataset combination of FC6, FC7 and FC8 layer representation give the best result, we can conclude that simple concatenation of multiple ConvNet layer representation does not always work better because of noisy and redundant features. These noisy and redundant features make the feature space more complex and give linear SVM a hard time to draw a decision boundary. <Table 1> result also matches our theory in section 3. Since SVM does not select features, we should expect little improvement or even worse performance when using multiple ConvNet layer representations against individual ConvNet layer representation.

To prove our hypothesis in section 3, we perform PCA to make feature selection and train SVM classifiers for each ConvNet layer representations (FC6, FC7, and FC8) and concatenated ConvNet layer representations (FC6-FC7-FC8). First three rows, FC6, FC7, and FC8 are single ConvNet layer representations. The last row (FC6-FC7-FC8) corresponds to the concatenation of multiple ConvNet layer features which is the representation that we proposed. As we can see from the <Table 2>, multiple ConvNet layers representation combined with feature selection is more potent than individual ConvNet layer representations. For instance, for Caltech-256, our proposed method achieves 1.7% higher accuracy (75.6%) than best individual layer representation (73.9%). For VOC07, our proposed method achieves 3.9% higher accuracy (73.1%), better than best individual layer representation (69.2%). Lastly, for the SUN397 dataset, our proposed method achieves 3.5% higher accuracy (52.2%) than best individual layer representation (48.7%). The most crucial procedure here is feature selection that was done by PCA. Recall that in <Table 1>, we obtained multiple ConvNet layer representation by merely concatenating three individual layer representations. This representation

〈Table 1〉 Linear SVM classifier accuracy results without PCA (%)

	Caltech-256	OC07	SUN397
FC6	71.4	69.1	**49.4**
FC7	**73.5**	70.6	47.3
FC8	72.6	70.2	45.2
FC6-FC7-FC8	73.2	**71.7**	48.7

〈Table 2〉 Trained SVM classifier accuracy results with PCA (%)

	Caltech-256	VOC07	SUN397
FC6	69.6	65.6	48.4
FC7	73.9	69	48.7
FC8	71.2	69.2	42.5
FC6-FC7-FC8	**75.6**	**73.1**	**52.2**

〈Table 3〉 FC6-FC7-FC8 as a feature extractor

	Caltech-256	VOC07	SUN397
SVM	73.2	71.7	48.7
AdaBoost (Jumabek et al., 2016)	72.8	71	49.1
SVM with PCA	**75.6**	**73.1**	**52.2**

contains a lot of noise and redundant features. However, when we perform PCA on those activation features, the best feature will be selected and that contribute to better classification. The reason is, as we explained in section 3, every individual ConvNet layers have different kind of feature extractors that are good at distinguishing the distinct part of specific classes. For instance, FC6 is good at distinguishing cat's eye, and FC7 is good at distinguishing cat's beard. Thus, when we combine these robust features, we will get better classification accuracy. That means, when we concatenate multiple ConvNet layer representations, we also combine multiple features that are robust for most classes to get a correct prediction. To get those robust features, we need to perform feature selection (PCA) before training. If we do not perform feature selection, like in <Table 1>, accuracy might be decreased because these combined features are very noisy and redundant.

Here in <Table 3>, we also compared our method with previous work (Jumabek et al., 2016) and plain SVM without PCA. In previous work, (Jumabek et al., 2016) use AdaBoost as feature selector. However, the problem with AdaBoost is it selects features implicitly on each training iteration and makes AdaBoost very slow at training stage.

The focal point of our work is showing the importance of feature selection from multiple ConvNet layer representation. We demonstrated the better performance of multiple ConvNet layer representation to facing against individual layer

representation by applying SVM as a classifier with PCA which makes feature selection (<Table 2>). In order to achieve superior performance, we need feature selection with multiple ConvNet layer representation. We showed SVM classifier without feature selection it would not achieve improvement in performance (<Table 1>).

5. Conclusions and Future Work

We proposed the idea of enriching image representation by combining features from multiple ConvNet layers. We claimed that using combined multiple ConvNet layers as a feature extractor can obtain better results than the single ConvNet layer. However, combined features contain noisy and redundant information, and that makes feature space more complex and hard time for linear SVM classifier to draw a decision boundary. Therefore, we need to select useful features from concatenated ConvNet representations. We introduced PCA feature selection as a solution for this. Moreover, we validated our idea through experiments by performing PCA feature selection with SVM classifier for both individual ConvNet layer representations and multiple ConvNet layers representation. Our proposed method achieved 1.7%, 3.9% and 3.5% accuracy improvement on Caltech-256, VOC07 and SUN397 dataset respectively compare to best individual ConvNet layer representation, which gave the superior performance that is achieved mainly by the feature selection procedure. To show the importance of feature selection, we also conducted another experiment without PCA. As expected using multiple ConvNet layer representation without feature selection, it gave little improvement and even worse performance. From the experimental results, we can conclude that multiple ConvNet layer representation performs better with feature selection. That validates our statement of the importance of feature selection.

Transfer learning method could be applied to variety of electronic products such as artificial intelligence refrigerators, home assistant, and auto dust cleaner. Therefore, we can apply multiple ConvNet layer as feature extractor for many kind of business applications such as stock exchange prediction, or customer classification for advertising market.

In our future work, we will estimate a more sophisticated approach to feature selection. In this paper, we used feature selection of PCA. Our method of selecting features using PCA as feature selector is one of the first steps. We assume there could be a lot more development by finding a better feature selector. Moreover, we will try deeper ConvNet model such as VGGNets and ResNet for feature extractor, and train different classifiers.

References

Abdi, H. and L. J. Williams, “Principal component analysis,” *Journal of Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 2,

No. 4(2010), 433~459.

Azizpour, H., A. Razavian, J. Sullivanm A. Make and S. Carlsson, "Factors of Transferability for a Generic ConvNet Representation," *IEEE*, 2014.

Donahue, J., Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv*: 1310.1531, 2013.

Everingham, M., S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, Vol. 111, No. 1(2015), 98~136.

Fukishima, K., "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, Vol. 36, No. 4(1990), 192~202.

Girshick, R., J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

Griffin, G., A. Holub and P. Perona, "Caltech-256 object category dataset," *California Institute of Technology*, 2017.

Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, 2014.

Jumabek, A., G. Myeong Hyeon, K. Seunghyun and J. Geun-Sik "Transfer Learning Based on AdaBoost for Feature Selection from Multiple ConvNet Layer Features", *Korea information processing society*, Vol. 23, No.1(2016), 633~635.

Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural netwokrs," in Advances in neural information processing systems, 2012.

Krizhevsky, A. and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, 2009.

Lee, J.-s., and H. . Ahn, "A Study on the Prediction Model of Stock Price Index Trend based on GA-MSVM that Simultaneously Optimizes Feature and Instance Selection", Journal of Intelligence and Information Systems, Vol. 23, No. 4 (2017), 147~168.

LeCun, Y., L. Bottou, U. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, Vol. 86, No. 11(1998), 2278~2324.

Oquab, M., L. Bottou, I. Laptev and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

Razavian, A., H. Azizpour, J. Sullivan and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A.

Khosla, M. Bernstein and others, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, Vol. 115, No. 3(2015), 211~252.

Schapire, R. E. and Y. Singer, “Improved boosting algorithms using confidence-rated predictions,” *Machine learning*, Vol. 7, No. 3 (1999), 297~226.

Song, J. H., H. S. Choi, and S. W. Kim, "A Study on Commodity Asset Investment Model Based on Machine Learning Technique", Journal of Intelligence and Information Systems, Vol. 23, No. 4 (2017), 127~146.

Sukjae, C., L. Jungwon, and K. Ohbyung, "Financial Fraud Detection using Text Mining Analysis against Municipal Cybercriminality", Journal of Intelligence and Information Systems, Vol. 23, No. 3 (2017), 119~138.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, “Going deeper with convolutions,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015), 1~9.

Xiao, J., K. A. Ehinger, J. Hays, A. Torralba and A. Olivia, “Sun database: Exploring a large collection of scene categories,” *International Journal of Computer Vision*, (2014), 1~20.

Zeiler, M. D. and R. Fergus, “Visualizing and understanding convolutional networks,” in *Computer Vision--ECCV 2014*, 2014.

국문요약

전이학습 기반 다중 컨볼류션 신경망 레이어의 활성화 특징과 주성분 분석을 이용한 이미지 분류 방법

바트후 뱜바자브* · 주마벡 알리하노브* · 팡양* · 고승현* · 조근식**

Convolutional Neural Network (ConvNet)은 시각적 특징의 계층 구조를 분석하고 학습할 수 있는 대표적인 심층 신경망이다. 첫 번째 신경망 모델인 Neocognitron은 80 년대에 처음 소개되었다. 당시 신경망은 대규모 데이터 집합과 계산 능력이 부족하여 학계와 산업계에서 널리 사용되지 않았다. 그러나 2012년 Krizhevsky는 ImageNet ILSVRC (Large Scale Visual Recognition Challenge) 에서 심층 신경망을 사용하여 시각적 인식 문제를 획기적으로 해결하였고 그로 인해 신경망에 대한 사람들의 관심을 다시 불러 일으켰다. 이미지넷 첼린지에서 제공하는 다양한 이미지 데이터와 병렬 컴퓨팅 하드웨어 (GPU)의 발전이 Krizhevsky의 승리의 주요 요인이었다. 그러므로 최근의 딥 컨볼루션 신경망의 성공을 병렬 계산을 위한 GPU의 출현과 더불어 ImageNet과 같은 대규모 이미지 데이터의 가용성으로 정의 할 수 있다. 그러나 이러한 요소는 많은 도메인에서 병목 현상이 될 수 있다. 대부분의 도메인에서 ConvNet을 교육하기 위해 대규모 데이터를 수집하려면 많은 노력이 필요하다. 대규모 데이터를 보유하고 있어도 처음부터 ConvNet을 교육하려면 많은 자원과 시간이 소요된다. 이와 같은 문제점은 전이 학습을 사용하면 해결할 수 있다. 전이 학습은 지식을 원본 도메인에서 새 도메인으로 전이하는 방법이다. 전이학습에는 주요한 두 가지 케이스가 있다. 첫 번째는 고정된 특징점 추출기로서의 ConvNet이고, 두 번째는 새 데이터에서 ConvNet을 fine-tuning 하는 것이다. 첫 번째 경우, 사전 훈련 된 ConvNet (예 : ImageNet)을 사용하여 ConvNet을 통해 이미지의 피드포워드 활성화를 계산하고 특정 레이어에서 활성화 특징점을 추출한다. 두 번째 경우에는 새 데이터에서 ConvNet 분류기를 교체하고 재교육을 한 후에 사전 훈련된 네트워크의 가중치를 백프로퍼게이션으로 fine-tuning 한다. 이 논문에서는 고정된 특징점 추출기를 여러 개의 ConvNet 레이어를 사용하는 것에 중점을 두었다. 그러나 여러 ConvNet 레이어에서 직접 추출된 차원적 복잡성을 가진 특징점을 적용하는 것은 여전히 어려운 문제이다. 우리는 여러 ConvNet 레이어에서 추출한 특징점이 이미지의 다른 특성을 처리한다는 것을 발견했다. 즉,

* 인하대학교 컴퓨터공학과
** 교신저자 : 조근식
인하대학교 컴퓨터공학과
#1209, Hi-tech center, Inha University, 100 Inha-ro, Yonghyun-dong, Nam-gu, Incheon 402-751, Korea
Tel: +82-32-860-7447, Fax: +82-32-875-5863, E-mail: gsjo@inha.ac.kr

여러 ConvNet 레이어의 최적의 조합을 찾으면 더 나은 특징점을 얻을 수 있다. 위의 발견을 토대로 이 논문에서는 단일 ConvNet 계층의 특징점 대신에 전이 학습을 위해 여러 ConvNet 계층의 특징점을 사용하도록 제안한다. 본 논문에서 제안하는 방법은 크게 세단계로 이루어져 있다. 먼저 이미지 데이터셋의 이미지를 ConvNet의 입력으로 넣으면 해당 이미지가 사전 훈련된 AlexNet으로 피드포워드 되고 3개의 fully-connected 레이어의 활성화 틀징점이 추출된다. 둘째, 3개의 ConvNet 레이어의 활성화 특징점을 연결하여 여러 개의 ConvNet 레이어의 특징점을 얻는다. 레이어의 활성화 특징점을 연결을 하는 이유는 더 많은 이미지 정보를 얻기 위해서이다. 동일한 이미지를 사용한 3개의 fully-connected 레이어의 특징점이 연결되면 결과 이미지의 특징점의 차원은 4096 + 4096 + 1000이 된다. 그러나 여러 ConvNet 레이어에서 추출 된 특징점은 동일한 ConvNet에서 추출되므로 특징점이 중복되거나 노이즈를 갖는다. 따라서 세 번째 단계로 PCA (Principal Component Analysis)를 사용하여 교육 단계 전에 주요 특징점을 선택한다. 뚜렷한 특징이 얻어지면, 분류기는 이미지를 보다 정확하게 분류 할 수 있고, 전이 학습의 성능을 향상시킬 수 있다. 제안된 방법을 평가하기 위해 특징점 선택 및 차원축소를 위해 PCA를 사용하여 여러 ConvNet 레이어의 특징점과 단일 ConvNet 레이어의 특징점을 비교하고 3개의 표준 데이터 (Caltech-256, VOC07 및 SUN397)로 실험을 수행했다. 실험결과 제안된 방법은 Caltech-256 데이터의 FC7 레이어로 73.9 %의 정확도를 얻었을 때와 비교하여 75.6 %의 정확도를 보였고 VOC07 데이터의 FC8 레이어로 얻은 69.2 %의 정확도와 비교하여 73.1 %의 정확도를 보였으며 SUN397 데이터의 FC7 레이어로 48.7%의 정확도를 얻었을 때와 비교하여 52.2%의 정확도를 보였다. 본 논문에 제안된 방법은 Caltech-256, VOC07 및 SUN397 데이터에서 각각 기존에 제안된 방법과 비교하여 2.8 %, 2.1 % 및 3.1 %의 성능 향상을 보였다.

주제어 : 딥러닝, 전이 학습, 고정 특징점 추출기, 활성화 특징점, 특징점 선택, 이미지 분류

논문접수일 : 2017년 12월 5일 논문수정일 : 2018년 2월 5일 게재확정일 : 2018년 2월 27일
원고유형 : 일반논문 교신저자 : 조근식

저 자 소 개

Batkhuu Byambajav

Received a B.S degree of Information Technology, Software Engineering from National University of Mongolia, Ulaanbaatar, Mongolia, in 2013. Now, he is pursuing a M.S degree in Computer Engineering in Inha University, Incheon, Korea and expected to graduate in February 2018. His research interests include Deep Learning, Machine Learning, and Image Classification.

Jumabek Alikhanov

Received a B.S. degree in Computer Engineering from Tashkent University of Information Technologies (TUIT), the Republic of Uzbekistan, in 2014, and a M.S. degree in Computer Engineering from Inha University, Korea in 2016. His research interests include Machine Learning, Artificial Intelligence, Computer Vision, and Deep Learning.

Yang Fang

Received a B.S. degree in Computer Engineering from Chongqing University of Posts and Telecommunications, China, in 2014. He is currently a Ph.D. Candidate in Computer Engineering of Inha University, Korea. His research interests include Artificial Intelligence, Computer Vision, and Deep Learning.

Seunghyun Ko

Received a B.S. degree in Computer Engineering from Texas A&M University, USA, in 2011, and a M.S. degree in Computer Engineering from Inha University, Korea in 2016. He is currently a Ph.D. Candidate in Computer Engineering of Inha University, Korea. His research interests include Artificial Intelligence, Computer Vision, and Deep Learning.

Geun-Sik Jo

Is a Professor in Computer and Information Engineering, Inha University, Korea. He received the B.S. degree in Computer Science from Inha University in 1982. He received the M.S. and the Ph.D. degrees in Computer Science from City University of New York in 1985 and 1991, respectively. He has been the General Chair and/or Technical Program Chair of more than 20 international conferences and workshops on artificial intelligence, knowledge management, and semantic applications. His research interests include knowledge-based scheduling, ontology, semantic Web, intelligent E-Commerce, constraint-directed scheduling, knowledge-based systems, decision support systems, and intelligent agents. He has authored and coauthored five books and more than 300 publications.