Convolutional Neural Network and the Recognition of Vehicle Types

Dewen Seng¹,²*, Bin Lin¹,², Jing Chen¹,²

ABSTRACT
In machine learning, a convolutional neural network (ConvNet) is a class of deep, feed-forward artificial neural networks. Featured by low computing load and fast convergence, the network has been successfully applied to pattern recognition. This paper gives a detailed introduction to the structure, working principles and advantages of ConvNet, and applies it to the recognition of vehicle types. In reference to previous research, two deep neural networks were created, namely VGG 16 and AlexNet. The experimental results show that our methods have performed well in vehicle classification in complex background images.

Key Words: Convolutional Neural Network (ConvNet), Recognition Algorithm, Pattern Recognition, Pooling Layer, Vehicle Recognition

DOI Number: 10.14704/nq.2018.16.6.1641

Introduction
In a neural network, each individual neuron is a computing unit and, in essence, a function. Below is the diagram of a neuron:

\[
 h_{w,b}(x) = f(W^T x) = f\left(\sum_{i=1}^{3} W_i x_i + b\right)
\]  (1)

Figure 1. The diagram of a neuron

In the diagram, there are 3 inputs, namely \(x_1\), \(x_2\), \(x_3\), and +1 is an offset value. The corresponding formula is as follows:

The unit is also known as the logistic regression model with \(f(\cdot)\) being the activation function. Here, the activation function is a sigmoid function. Figure 2 shows the sigmoid function of the image.

Figure 2. Sigmoid function

The threshold of activation is 0.5. The output of the sigmoid function varies between 0

Corresponding author: Dewen Seng
Address: ¹Key Laboratory of Complex Systems Modeling and Simulation Ministry of Education, Hangzhou Dianzi University, Hangzhou 310018, China; ²School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China.
E-mail: *sengdw@163.com

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 6 March 2018; Accepted: 3 May 2018
and 1. The neurons are either in the "activated" or "non-activated" state. The activated neurons can transmit signals, while the non-activated ones cannot transmit signals. The division is simulated in the function design of artificial neural networks application.

**Network structure of convolutional neural network (ConvNet)**

The ConvNet is an artificial neural network heatedly discussed in the fields of image recognition and speech analysis. The weight sharing structure reduces the model complexity and the number of weights, making the ConvNet similar to the biological neural network. The effect of weight sharing is particularly obvious when the input is a multidimensional image. The image can be used directly in ConvNet without going through the complex process (e.g. feature extraction and data reconstruction) of the traditional recognition algorithm. Besides, the network uses a variation of multilayer perceptrons to recognize 2D shapes and special designs. This kind of structure can withstand translation, scaling, tilting or deformation.

The ConvNet mainly has the following functions:

1. **Feature extraction**
   Each neuron takes its synaptic inputs from a local receptive field, called a convolutional kernel from the previous layer, thereby forcing it to extract local features. Once a feature has been extracted, its location becomes less important as long as its position relative to other features is approximately preserved.

2. **Feature mapping**
   Each computing layer of the network is composed of many planar feature maps. In the same plane, the neurons share the same set of synaptic weights. Such this structure stays immune to translation, and reduces the number of free parameters by weight sharing.

3. **Sub-sampling**
   Each convolutional layer is supported by a local average and sub-sampling calculation layer. This support layer can reduce the resolution of feature map output to translation and other forms of deformation.

Figure 3 illustrates the filtering process of each input image through three trained filters and the assignment of additive bias by convolution. The convolution in the C1 layer produces three feature maps. Based on the sum of four pixel values in each feature map and the additive biases, the feature maps of the three S2-level feature maps are obtained by a sigmoid function. These maps are then filtered to get the C3 layer. Then, the same process is repeated to generate S4 from S2. Finally, these pixel values were grated and combined into a vector, and inputted into the traditional neural network to obtain the output.

**ConvNet**

Typically, the convolutional layers of the ConvNet are interspersed with sub-sampling layers to reduce computation time and achieve spatial and measurement invariance. To maintain specificity, it is desirable to have a small sub-sampling factor. This structure is commonly seen in mammalian visual cortex and relevant models (Geismann and Schneider G., 2008; Akbari and Rezaei, 2009; Hasler et al., 2009). According to the latest findings in auditory neuroscience, similar structures have been discovered in the auditory cortex of a number of animals (Pfister et al., 2015; Yeung et al., 2005; Shamma, 2013; Huang et al., 2017).

1. **Local perception**
   In the ConvNet, there are two ways to reduce the number of parameters. The first approach is based on local receptive field. It is generally believed that the human perceives the outside from local to global (Olshausen and Field, 1996). In an image of a space, the near pixels are closer correlated than those in the distance. When the image is processed by an artificial neural
network, it is not necessary to perceive the global pixels. Instead, the neurons only have to perceive the local pixels so that the overall information can be derived from the local information at the higher level. The local-connected network is also inspired by the structure of the visual system. The neurons in the visual cortex only receive the local information. Figure 4 shows full-connected network and local-connected network.

On the right side of Figure 4, if each neuron is connected to 10×10 pixels, then the weight data is 10,000×10,000 parameters, about 1/1,000 of the original data. The 10×10 pixel values correspond to 10×10 parameters. The above step is called a convolution operation.

Figure 4. Fully connected neural net and Locally connected neural net

(2) Weight sharing
Despite the reduction process, the parameter size is still too huge to use. This calls for the further size reduction by weight sharing. In the said local-connected network, there are a total of 1,000,000 neurons, each of which corresponds to 100 parameters. If the 100 parameters are constant from neuron to neuron, then the number of parameters is equivalent to 100.

The convolution operation of these 100 parameters can be regarded as a way to extract features. Independent of position, the feature extraction operation works on the principle that the statistical features of a part of the image is the same as those of the other parts. This means the local features can be applied globally, i.e. the same features can be used across the image (Ström, 1997).

(3) Convolution kernel
The two reduction processes leave only 100 parameters, indicating that there is only one 100×100 convolution kernel. It is obvious that the features cannot be fully extracted with only one kernel. Hence, multiple kernels should be introduced (e.g. 32 kernels) to learn even more types of features. The multi-kernel scenario is illustrated in Figure 5.

On the right side of Figure 5, the different convolution kernels are in different colours. Each kernel generates an image into another image. For example, the two images generated by two different kernels can be seen as one image of different channels. Note that the number of images is equal to the times of feature mapping by neurons. The larger the image, the wider is the gap between the number of neurons and parameters.

Figure 5. Locally connected neural net and Convolutional net

Convolution
Convolution is a very important and complex mathematical operation. For simplicity, only the discrete form of convolution is introduced here. The convolution operation on the image is actually a filtering process (Won et al., 1997). Let \( I = f(x, y) \) be an image. Then, the basic expression of convolution is as follows:

\[
f(x, y) \times w(x, y) = \sum_{s=-a}^{b} \sum_{t=-b}^{a} w(s, t) f(x-s, y-t)
\]

(2)

Where \( f(x, y) \) means image \( I \) is in the X row and Y column in the grey coordinate system; \( w(x, y) \) is the convolution kernel (aka. filter, response function etc.); \( a \) and \( b \) define a convolution kernel of the size \( w(x, y) \).

According to the above formula, the convolution actually provides a template that slides on the image, and aligns its centre to each pixel in the image (Serre et al., 2002; Arribas et al., 1999); then, all the pixels covered by the template are weighted, and the results are taken as the response of the convolution kernel in the image.
ConvNet algorithm
It is found that the visual cortexes contain neurons that individually respond to small regions of the visual field. Provided the eyes are not moving, the region of visual space within which visual stimuli affect the firing of a single neuron is known as its receptive field (Melorose et al., 2015; Wang et al., 2016). Two basic visual cell types are identified in the brain: the simple cells, whose output is maximized by straight edges having particular orientations within their receptive field, and the complex cells, which have larger receptive fields, whose output is insensitive to the exact position of the edges in the field (Sun and Feng, 2007).

Sparse connection
In a back-propagation (BP) neural network, each layer of neuron node is a linear one-dimensional array structure, and the nodes between different layers are fully connected. In the ConvNet, the interlayer nodes are no longer fully connected. Instead, these nodes are local-connected based on the spatial correlation of nodes in adjacent layers (Szarvas et al., 2005). In this research paper, the convolutional neural network has a local-connected structure.

Weight sharing
In the ConvNet, each filter on the convolution layer acts repeatedly on the entire receptive field. The convolution results constitute the feature map of the input image, laying the basis for the extraction of local features. The filters share the same parameters, including the weight matrix and additive bias.

Maximum pool sampling
Another important concept about ConvNet is the nonlinear down-sampling method called maximum pool sampling. The image features acquired by convolution make it possible to extract the training data for classifier. However, the data extraction process demands a great amount of computation. For example, suppose there are a 48×48-pixel image and a convolutional layer of 300 4×4 filters. Then, each kernel has to handle 300 features in (48-4+1)×(48-4+1) dimensions. Hence, each sample will convolve feature vectors in 45×45×300=607,500 dimensions.

Feedforward pass
The following derivation considers the squared error loss function. For a problem with c classes and N training samples, the squared error is expressed as:

\[ E^N = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{c} (t^n_k - y^n_k)^2 \]  

(3)

Where \( t^n_k \) is the k-th dimension of target of the n-th image; \( y^n_k \) is the value of the k-th output layer unit in response to the n-th input image. For a multiclass classification problem, the targets are typically organized as a "one-of-c" code where the k-th element of \( t^n \) is positive if the image \( x^n \) belongs to class k. The rest entries of \( t^n \) will be either zero or negative depending on the choice of the output activation function.

Since the error of the entire dataset is the sum of the individual error of each image, the BP of the n-th image is taken into account:

\[ E^N = \frac{1}{2} \sum_{k=1}^{c} (t^n_k - y^n_k)^2 = \frac{1}{2} \| t^n - y^n \|^2 \]  

(4)

Gradient calculation of convolution layer
This section deals with the BP updates for convolutional layers in a network. In a
convolution layer, the feature maps of the previous layer are convolved with learnable kernels and put through the activation function to form the output feature maps. Each output map may involve the convolution of multiple input maps. The relationship is expressed as:

\[ x_j^h = f \left( \sum_{i \in M_j} x_i^{h-1} * k_{ij}^h + b_j^h \right) \]  

(5)

Where \( M_j \) means the selection of input maps. If it is implemented in Matlab, the convolution belongs to the handling of valid border conditions. Common choices of input maps include all pairs or all triplets. Below is a discussion of the learning of combinations. Each output map is given an additive bias \( b \). However, for a particular output map, the input maps are convolved with distinct kernels. In other words, if output map \( j \) and map \( k \) both sum over input map \( i \), then the kernels applied to map \( i \) are different from those to maps \( j \) and \( k \).

In order to calculate the error signal of layer \( \ell \) more effectively, it is necessary to sample the error signal of the next layer, ensure that the signal has the same size with that of the convolution layer, and up-sample the error signal based on the partial derivatives of the activation function in layer \( h \). Whereas all the weights in the down-sampling layer are equal to \( \beta \) (a constant), \( \delta^h \) can be obtained by expanding the result of the previous step to \( \beta \) times. The above steps should be repeated to calculate the error signal of each layer in image \( j \), and the relative reduction of the sampling layer can be expressed as:

\[ \delta_j^h = \beta_j^{h-1} \left( f \left( u_j^h \right) * up \left( \delta_j^{h-1} \right) \right) \]  

(6)

In light of the sensitivities for a given map, the bias gradient can be obtained through summation over all the entries in \( \delta_j^h \):

\[ \frac{\partial E}{\partial b_j} = \sum_{u,v} \left( \delta_j^h \right)_{uv} \]  

(7)

(2) Dataset size

The performance of data-driven models generally depends on the size of dataset. Although the ConvNet and other empirical models are applicable to dataset of any size, the training dataset should be large enough to cover all kinds of possible problems.

During the design of ConvNet, the dataset should contain three subsets: the training set, the test set, and the validation set. The training set, responsible for network weight adjustment in the training stage, must cover all the data in the problem domain. The test set helps to check the network classification of training data in the training process. The network structure should be adjusted or the training cycles should be increased according to the network performance of the test set. The validation set verifies the test results and determines the network structure based on the measured performance.

Data pre-processing

Data pre-processing techniques are often introduced to speed up the convergence of training algorithms, including but not limited to the removal of noise, the reduction of input data dimensionality and the deletion of irrelevant data. The data balance is essential to classification. In general, a training dataset should be distributed evenly like the labels of each category. The labelled data of each category in the training data is basically the same, so as to avoid the overemphasis some features in the classification table. In order to balance the dataset, the surplus data for classification should be removed and replaced with relatively scarce data. For instance, one can copy a portion of the

\[ x_j^h = f \left( \beta_j^{h} \text{down} \left( x_j^{h-1} \right) + b_j^h \right) \]  

(8)

Where \text{down} (·) is the down-sampling function. The function processes each N×N area in the input map, and ensures that the output map is \( n \) times smaller than the input map in any dimension. Each output map has its own multiplier deviation \( \beta \) and additive bias \( b \). With only \( \beta \) and \( b \) available, it is very difficult to calculate the error signal. It is assumed that both the upper layer and the lower layer are convolution layers. If the down-sampling layer is connected to the network, the error signal can be directly obtained through the BP algorithm.
scarce data and add random noise to the input data.

(1) Initialization of network weights
The initialization of network weights refers to the assignment of initial values to all connection weights in the network (including the threshold). If the initial weight vector lies in a relatively flat area of the error surface, the convergence rate of the network training may be very slow. Under normal conditions, the network connection weights and threshold are initialized in a relatively small range of zero-mean uniform distribution.

(2) Learning rate of BP algorithm
In the training process, a relatively fast learning rate \( \eta \) can speed up network training via weight adjustment, but it may cause the network to fluctuate frequently on the error surface and hinder the convergence of the training process. On the contrary, a relatively slow learning rate can drive the network towards the global optimum, but it cannot eliminate the chance of falling into the local optimum. In view of the strengths and weaknesses of different learning rates, an adaptive learning method has been proposed to adjust the training process with the training algorithm.

**Application of ConvNet in Vehicle Recognition**
Nowadays, vehicle recognition is a hot topic in the research on smart transport system. Owing to the following difficulties, however, the existing research results are rarely applied to the actual practice.

First, the vehicle images taken from the road are complicated by the background (e.g. roadside trees and buildings), the barriers between vehicles, and the noises of the vehicles (e.g. ads stickers, occupants, etc.).

Second, the vehicle images may suffer from distortion due to the camera position and vehicle type, adding to the difficulty of image pre-processing.

Deformed or distorted by the camera position

Third, the quality of vehicle images is affected by weather. Different weather conditions have varied degrees of impact on image quality. Most of the existing images were taken on sunny days, failing to reflect the vehicle conditions in other weather conditions.

Fourth, the fast update of vehicles causes rapid changes to the shape and other features, leading to poor adaptability of many algorithms.

These problems make it difficult to implement vehicle recognition in the management of intelligent transport system. In order to realize the implementation, much work needs to be done in the future.

**Introduction to vehicle recognition system**
The vehicle recognition system is an important part of highway toll system. The system integrates digital image processing, pattern recognition, computer vision, electronic technology and system engineering into an organic whole. The identification of vehicles based on specific time and place lays a solid foundation for traffic management, scheduling and charging.

The vehicle recognition system can classify vehicles by brand, size or type. There is no uniform classification standard in China.

**Establishment of model image database**
Speed and accuracy are two key parameters of an image recognitions system. The recognition speed mainly depends on the duration of feature extraction and test. The recognition accuracy, as a determinant of the recognition effect, refers to the proportion of correctly identified samples in the total number of samples. To create an optimal convolution network, the speed index in this research is determined by following aspects:

1. ConvNet network training duration;
2. Classifier training duration;
3. Feature extraction duration;
4. Classification recognition duration;
5. Feature dimensions.

The feature dimensions were extracted to ensure the correctness of the final classification.

The feature extraction duration and classification recognition duration are the main reference indices, while ConvNet network training duration, classifier training duration and feature dimensions are secondary indices.

Once it was established and trained, the ConvNet feature extraction and SVM classification training model could be applied to the same class of datasets. There was no need for repeated operations. There might be a slight difference in the features of each particular structure, because the network supports autonomous learning and feature extraction.
Ranging from include vehicle detection, feature extraction and selection to pattern recognition, the traditional vehicle recognition techniques face many difficulties. First is the segmentation of the target vehicle from the complex background. Figure 8 show some vehicle images taken from various angles. From top to bottom, there are SUVs, pick-up trucks, vans and cars. The target vehicle segmentation quality hinges on the vehicle classification. Then come the auto selection of typical features and conversion of these features into effective parameters. Third is the correct selection of feature parameters and proper classifier design. In this paper, our neural network takes the original image as the input, treats the original data through the convolutional layer, and processes the fully-connected layer by the softmax layer. In this way, the image segmentation and manual feature extraction were eliminated from the procedure.

**Table 1. AlexNet network configuration**

<table>
<thead>
<tr>
<th>Receptive Field</th>
<th>Step</th>
<th>Fill</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input 224*224 RGB image</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv 11×11</td>
<td>4</td>
<td>0</td>
<td>96</td>
</tr>
<tr>
<td>Max-Pooling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv 5×5</td>
<td>1</td>
<td>2</td>
<td>256</td>
</tr>
<tr>
<td>Max-Pooling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>384</td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>384</td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>256</td>
</tr>
<tr>
<td>Max-Pooling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FC</td>
<td>—</td>
<td>—</td>
<td>4096</td>
</tr>
<tr>
<td>FC</td>
<td>—</td>
<td>—</td>
<td>4096</td>
</tr>
<tr>
<td>FC</td>
<td>—</td>
<td>—</td>
<td>4096</td>
</tr>
<tr>
<td>Softmax</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2. VGG16 network configuration**

<table>
<thead>
<tr>
<th>Receptive Field</th>
<th>Step</th>
<th>Fill</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input 224×224 RGB image</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>Max-Pooling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>Max-Pooling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>256</td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>256</td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>256</td>
</tr>
<tr>
<td>Max-Pooling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>Conv 3×3</td>
<td>1</td>
<td>1</td>
<td>512</td>
</tr>
<tr>
<td>Max-Pooling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FC</td>
<td>—</td>
<td>—</td>
<td>4096</td>
</tr>
<tr>
<td>FC</td>
<td>—</td>
<td>—</td>
<td>4096</td>
</tr>
<tr>
<td>FC</td>
<td>—</td>
<td>—</td>
<td>4096</td>
</tr>
</tbody>
</table>

In the above tables, Conv is the convolutional layer, FC is a fully-connected layer, Pooling is the pool layer, and Softmax is the classification layer.

(1) Dataset
Our dataset contains four types of vehicles: SUVs, pick-up trucks, vans and cars. The training set and test set cover 998 and 248 images, respectively. In the dataset, the images were taken from all angles, and featured by nonuniform sizes and complex backgrounds. The proportion of target vehicle in the image varied from case to case, adding to the difficulty in vehicle recognition. To maintain a consistent input size, the original images were adjusted to a uniform size of 256 * 256 * 3. Then, the mean image in the 3 RGB channels were derived, and the input data were subject to the zero-mean standard processing. In the network training test, 224 * 224 *3 samples were selected as the input.

(2) Network structure
The deep neural network VGG16 was adopted for this research. The network consists of 5 stacked ConvNets, each of which has 3 fully-connected layers, 1 softmax layer and a max-pooling layer. After convolution and pooling, the output of the 3 fully-connected layers was taken as the input of the softmax layer, and the result of the final layer was the classification results of vehicle types. Then, a nonlinear ReLu layer was introduced to the network to process the output of the fully-connected layers. The new layer greatly
shortened the duration of network training. Moreover, the regularization method Dropout was adopted in the network to prevent over-fitting in the fully connected layers.

Then, another network AlexNet was also employed in this research. Similar to the VGG, this network also encompasses five stacked ConvNets, each of which has 3 fully-connected layers, 1 softmax layer and a max-pooling layer. The nonlinear ReLU layer was also introduced to reduce the chance of over-fitting.

Table 3. Classification results of VGG16, AlexNet and KNN

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>97.6%</td>
</tr>
<tr>
<td>AlexNet</td>
<td>93.0%</td>
</tr>
<tr>
<td>KNN</td>
<td>50.4%</td>
</tr>
</tbody>
</table>

(3) Experiment design

Our deep neural networks VGG16 and AlexNet were built in the Caffe framework, and ran on a NVIDIA GeForce GTX TITAN X workstation. The two networks managed to achieve very high operation efficiency. It takes only 2 hours to train a single network, and merely 0.2 second to test an image. For comparison, the classic classification algorithm KNN was introduced for vehicle type classification.

Conclusions

According to the experimental results in Table 2, VGG16 had the best performance in vehicle classification in complex background images with an accuracy of 97.6%, followed in descending order by AlexNet (93.0%), and the KNN algorithm (50.4%).

In terms of the images on individual vehicles, the VGG16 successfully recognized all van images. The network made wrong classification of three SUV models. The first misclassification is attributed to the high similarity between the model structure and the pickup truck. The second model was incorrectly identified because the red colour in half of the vehicle coincides with the colour of pickup truck models. The third misclassification is due to the presence of other cars in the background. Besides, the image only contains the front of the vehicle, making it difficult to separate the model from the background information.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61473108), the Public Projects of Zhejiang Province (No. 2013C33082), Zhejiang Provincial Natural Science Foundation (No. LY15F020038 and No. LQ13F020005) and the research foundation of the Education Department of Zhejiang Province (No. Y201430884).

References