Regression Convolutional Network for Vanishing Point Detection

Yan Shuai1,2, Yan Tiantian 1,2, Yang Guodong2, Liang Zize2

1. University of Chinese Academy of Sciences, Beijing 100049

2. State Key Laboratory of Management and Control for Complex Systems, Institute of Automation of, Chinese Academy of Sciences, Beijing 100190

E-mail: yanshuai2015@ia.ac.cn, yantiantian2015@ia.ac.cn, guodong.yang@ia.ac.cn, zize.liang@ia.ac.cn

Abstract: This paper presents a detection method for estimation of vanishing point position with designed regression convolutional neural network. Due to the deep structures of convolutional networks, global high-level features are extracted from the whole image, which helps to locate the vanishing point. In this paper, we provide a new structure of regression neural network based on AlexNet. The structure consists of five convolutional layers, four fully connected layers, Tanh activation function and regression loss function. We feed the neural net with a small number of training dataset and the result proves that this method is adaptable. Compare to classical method, deep learning is more effective on blurred pictures and complex circumstances.

Key Words: CNN, regression, vanishing point, AlexNet

1 INTRODUCTION

There are a lot of parallel lines in the artificial 3D scene. In the process of projection into a two-dimensional picture, the parallel lines of the three-dimensional space are mapped into straight lines that intersect at one point in a two-dimensional image, this point is the vanishing point. The vanishing point contains the direction information of the line, it is more robust compare to other features while providing a description and understanding of the environments. Therefore, vanishing point estimation is an important research topic in the field of computer vision and is widely used in robot navigation, 3D reconstruction, camera calibration and so on.

Currently, methods of vanishing point detection can be categorized as follows: first category, using the spatial transformation technique, the image is transformed into another limited space. Using spherical radial base function neural network to estimate the cluster parameter to solve the problem of vanishing point analysis improves the efficiency of the algorithm, reducing the impact of noise, but the performance of this method on nonparametric models is limited[1]. Gaussian mapping based vanishing point detection has a problem caused by uneven accumulator unit. According to the feature of man-made object, the paper [2] provides a length-preferential vector cross product vanishing point detection method by combining segment length with Gaussian mapping and vector cross production. This method get more reasonable vanishing points which suitable for description of man-made object. Second category, the location of the vanishing point is estimated using local features such as textures. For instance, the paper [3] propose a new method for detecting the vanishing point of unstructured road based on Haar texture, by using integral image technique, the complex response of Haar texture is fast calculated according to the carefully designed real and imaginary Haar templates. The algorithm in the paper [4] first uses Gabor filter to calculate the texture orientation of each pixel. Then, it determines whether or not to take part into the voting process according to the confidence level of the pixels. Finally it uses fast local voting algorithm to estimate the positions of vanishing points. The paper [5] use Gabor filter to estimate texture orientation and compute the confidence of pixels, a group of imaginary downward rays are founded, then the orientation consistency ratio weighted by both color difference and double-angle sine function is calculated, and the ray with maximum value is selected as the first road edge. This method is accurate and robust in rural road. The paper [6] classify four road types based on an abstracted holistic road feature. After classification, the lane or edges in a specific scale is chosen for the line-segment-detection or region segmentation, and vote the intersection point. The paper [7] searches for vanishing point candidates by multiple population genetic algorithm, then use a voting method obtains the value of the fitness function, lastly, a estimation method applies Gabor filter banks to estimate the local dominant orientation of the vanishing point candidate and its voters. Third category, use the straight line feature to compute the line intersection and then estimate the vanishing point position. The paper [8] raise a new vanishing point detection algorithm based on envelope of perpendicular and parallel lines is proposed for typical urban road and street image. The paper [9] proposed a method decomposed a 2d Hough parameter space into two 1D Hough parameter spaces. This method performs robust and accurate in outdoor artificial scene images. The paper [10] add a preview model parameter evaluation selection to the RANSAC algorithm and a significant increase in speed is shown and the results are same as RANSAC. With these
less adaptable methods, it is difficult to detect the vanishing point of a complex environment.

Deep learning is widely used in target detection, identification and regression tasks because of its adaptability and no need of artificial modeling. But there is few study of vanishing point detection based on deep learning method. The paper [11] follows a data-driven learning approach by training two popular convolutional neural networks, AlexNet and VGG. In this paper, 37,497 frames (resized to 300 × 300 pixels) is taken as input of the CNN. Every frame is annotated with one vanishing point. Suppose we divide each picture with n orthogonal grid (n = [10, 20, 30]), the vanishing point will fall somewhere in the grid. The cell that the vanishing point falls into is set to 1, other cells are set to 0. And then, the values of these cells are arranged as a row vector of 900 (n = 30) or 400 (n = 20) or 100 (n = 10) as the target vector. The output of the network is also designed as a row vector of the same length. The loss function is designed as follows.

\[
loss = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{C-1} \left\{ y_j^{(i)} \cdot \log[h_j(x^{(i)})] \right\} 
\]

Where N is the batch size; C is the number of output of the network. However, the accuracy of this method for detecting vanishing point in complex scenes still needs to be improved.

To solve the above problem, we propose a vanishing point detection method based on regression CNN, the method can be used to detect the vanishing point of complex environments. The rest of the paper is organized as follows. Section 2 briefly introduces the structures of AlexNet and the proposed network. Section 3 offers the implementation details, we modified the activation function and the loss function. Section 4 offers the experimental results on the vanishing point detection under multiple complex scenes, presents the comparison with the previous method and reports the analysis of the results. Finally, conclusions are drawn in Section 5.

2 NETWORK STRUCTURE

2.1 Network Structure Selection

First of all, the network structure needs to be deep, because the vanishing point is not a local information, but a result of integrating the entire picture information. Predicting the vanishing point requires a high level neural network. Depth neural networks can combine local features to form more abstract features. In the application of vanishing point detection, the first few convolutional layers can get the information of the edge. In the subsequent convolutional layers, this information is integrated into more abstract information. By further abstraction of more convolutional layers and fully connected layers, the relationship of the feature can be obtained. In addition, this feature extraction method is a kind of complex non-linear mapping thanks to deep non-linear network. This complex nonlinear mapping provides the possibility of vanishing point detection in an extremely complex and changing environment. Unlike traditional modeling methods, CNN can learn this non-linear relationship between the input images and the vanishing points through automatic learning of the training set.

2.2 Structure of AlexNet

The regression network for the detection of vanishing points is based on AlexNet, therefore, we first introduce the network structure of AlexNet. The overall structure of the AlexNet network is shown in Figure 1. The network contains eight layers that require weight optimization. These eight layers contains five convolutional layers (Conv1, Conv2, Conv3, Conv4, Conv5) and three fully connected layers (FC6, FC7, FC8). Besides, Conv1 and Conv2 is followed by normalization layers. Max-pooling layers is behind all normalization layers and the last convolutional layer. The rectified linear units (ReLUs) process the output data of all convolutional layers and fully connected layers. The first convolutional layer has 96 kernels with shape of 11 × 11 × 3 and stride of 4. (Stride is the distance between the receptive field centers of neighboring neurons in a kernel map). The second convolutional layer takes the output of the first convolutional layer as input. The second convolutional layer has 256 kernels with shape of 5 × 5 × 96. The third / fourth / fifth layers behind the second convolutional layer are connected in turn. The third convolutional layer has 384 kernels with shape of 3 × 3 × 256, the fourth convolutional layer has 384 kernels with shape of 3 × 3 × 384 and the fifth convolutional layer has 256 kernels with shape of 3 × 3 × 384. The fully connected layers FC6 / FC7 each have 4096 output neurons. The output of the eighth fully connected layer, that is, the output of the whole network has 1000 neurons which representing the classification of 1000 categories. The 1000 outputs are processed by Softmax to get the class probability. The network also uses the dropout layer to prevent overfitting. Because of extraction of high-level abstract features of the image, AlexNet performance very well in ImageNet.

2.3 The Network Structure of the Vanishing Point Detection
According to the characteristics of vanishing point detection, AlexNet was modified to the structure in Figure 2. The input of the network is a color picture with size of $224 \times 224$ in the training set. The two-dimensional coordinates of the vanishing point are manually annotated. The figures on each dimension are normalized to the range of 0-1. The coordinate will be $(0,0)$ at the top left corner of the image, and the coordinate will be $(1,0)$ at the top right corner of the image. The network consists of five convolution layers and four fully connected layers. The output of all convolutional layers and the first three fully connected layers are processed by the Tanh activation function. The first / second / fifth layer is followed by the max-pooling layer. First three fully connected layers are followed by dropout layer in order to prevent over-fitting. These layers are connected in turn, and the last layer provides the predictions of two-dimensional coordinates. Finally, update the parameters of the network through the loss function and the optimizer.

2.4 Implementation Details

The convolution layer is annotated by $C(s,n)$ in which $s$ is the width and height of the convolution kernel(also known as a filter) and $n$ is the number of output maps. The input is a matrix of size $(h,w,m)$ in which $m$ is the number of input maps and $h$ and $w$ are respectively the height and width of each map. Therefore, the convolutional layer annotated by $C(s,n)$ is defined as follows:

$$y_{i,j}^{(0)} = \tanh\left(\sum_{r=0}^{m-1} \sum_{k=0}^{n-1} x_{i+k,j+r}^{(0)} \cdot w_{k,j}^{(0)} + b_{i,j}^{(0)}\right)$$

Here, $x$ and $y$ are the input and output of this layer, $w$ is the weight, $b$ is the bias. Notice that we replace the original ReLU activation function of the AlexNet by the Tanh activation function. This is because the experimental results show that ReLU may lead to poor convergence because of abnormal weight. After replacement of Tanh, Although the initial situation will appear similar to ReLU, but after a period of time, the loss began to fall again, and eventually converge to a very small limit.

Pooling layers is annotated by $P(s)$ in which $s$ is the length and width of the pooling template. The $P(s)$ is formulated as

$$y_{i,j}^{(0)} = \max_{0\leq k \leq s} \{x_{i+k,j+s}\}$$

Behind the convolutional layers and the pooling layers are the fully connected layers. The classic AlexNet is used for classification, the application scenario of vanishing point requires us to change it to regression. We think this will increase the calculation work of fully connected layers. Therefore, we added another fully connected layer. But a network with a smaller number of fully connected layers may have a good performance as well. Fully connected layers is annotated by $F(n)$ with function

$$y_j = \tanh\left(\sum_{i=0}^{n-1} x_i \cdot w_{i,j} + b_j\right)$$

For $j = 0, \ldots, n-1$, where $n$ and $m$ are the numbers of neurons at the current layer and previous layer. After each fully connected layer, there is a dropout layer to prevent over-fitting.

The regression function is designed to be the square of Euclidean distance of the coordinates of the network output and the coordinates of the annotation on the image. The function is defined

$$loss = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{i=0}^{C-1} (y_{i}^{(0)} - h_{i}^{(0)}(x_{i}))^2$$

Where $N$ is the batch size; $C$,the number of output of the network, is set to be 2 since the vanishing point of the network output is a two-dimensional coordinate; $y$ represents the real position, $h_{i}^{(0)}(x_{i})$ is the true value of the x-coordinate of the i-th sample while $h_{i}^{(0)}(x_{i})$ is the true value of the y-coordinate of the i-th sample; the output of the network is $h$ in which $h_{i}^{(0)}(x_{i})$ is the y-coordinate predicted value of the i-th sample and $h_{i}^{(0)}(x_{i})$ is the y-coordinate predicted value of the i-th sample.

3 EXPERIMENTS & RESULTS

3.1 Training Data Collection

The original data is 640 images with a vanishing point. The main types of these images are straight road, straight railway, underground pipe gallery, indoor scenes, gallery and computer virtual picture. By rotating, flipping and cropping transforms, these original images are expanded to 5548 images. Then, 5548 images were resized into $224 \times 224$ pixel size images after which annotate the vanishing points. The vanishing pixel is divided by 224 to get a coordinate which is between 0 and 1. The network inputs are these 5548 pictures. Each batch contains 100 pictures. Coordinates which are between 0 and 1 are set to be the target values of regression. The training process trained 6000 epochs and took about 50 hours.

3.2 Experimental Environment

Table 1. Data Sheet of Training Environment

<table>
<thead>
<tr>
<th>Hardware</th>
<th>specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Xeon CPU E5-2630 v3</td>
</tr>
<tr>
<td>GPU</td>
<td>NVIDIA GeForce GTX TITAN X</td>
</tr>
<tr>
<td>Memory Capacity</td>
<td>64GB</td>
</tr>
<tr>
<td>Software</td>
<td>tensorflowyoutu-1.0.1</td>
</tr>
<tr>
<td>Operatin System</td>
<td>Linux ubuntu</td>
</tr>
</tbody>
</table>

The training process, which is to get the vanishing point detection model, runs on a server with the configuration showed in Table 1. The testing process is on the local computer, whose operating system is windows 10 and the version of tensorflow is tensorflow_cpu-1.0.1.
3.3 Results

In this section, we will show the convergence curve of the loss function, the detection results of the training set and the test set.

**Loss curve.** Loss curve is shown in Figure 3. The initial convergence rate is very fast, after about 100 batches, the loss value has fallen to around 0.1. The rapid decline at the beginning is because of the fact that all vanishing points are in the picture while the predictions are outside the picture. After the vanishing point predicted by the network falls into the image, in a fairly long period of time, the result of the predicted vanishing point is roughly equivalent to randomly throwing a point in the picture. The final process is the second decline of the loss curve. After this decline, the location of the predicted vanishing point is very close to the position of annotation.

![Fig 3. The loss curve of training. The stable value is around 0.0038.](image)

**Results of training dataset.** We randomly selected some images (Figure 4) in the training dataset to show the results of the training. The red dot in the picture is the predicted vanishing point while the blue dot is the vanishing point of annotation.

![Fig 4. The detection results of training dataset.](image)

**Results of test dataset.** The test dataset has 1000 images, which is different with the training dataset, some are shown in Figure 5. Our method is compared with classical method. This classical method firstly detects edges and extracts the line features. Then it gets the vanishing point by analysis the intersections of the straight lines. We chose some pictures with obvious straight lines to fit this method. And the parameters of this method is well-adjusted. The result showed in Figure 6, illustrates our proposed method performs better.

Additionally, we compared our method with a kind of classification CNN networks mentioned in the paper [11]. The result (Figure 7) show our approach is more robust and accurate.

![Fig 5. The detection results of test dataset.](image)

![Fig 6. The comparison between classical method and our method.](image)

![Fig 7. The comparison between the method based on the classification of the cnn network and our method.](image)

**Analysis.** We have selected some representative pictures to illustrate the adaptability of our method. These pictures represent the performance of this method in some special cases. A in Figure 8 shows that the method in this paper has a good adaptability to the rotating scene. B in Figure 8 shows that this method has a good performance in the dark environment. C in Figure 8 shows that the method in this paper has a good performance for irregular pavement. B and C show that the method in this paper does not require very sharp edge features, the gradient edge can be well detected. Therefore it performs well in the irregular pavement. D in Figure 8 shows that the method in this paper is not a simple
extraction of local features, and then analysis these simple features. The network may have found out the regulation of shrink and expand of one pattern. This is unreachable by the conventional method which simply processes information of feature. E in Figure 8 shows that in a picture that is stitched by multiple images, the network will find the most obvious vanishing point. The right in Figure 8 shows that the network also has a good performance in the environment with many interferences.

Fig 8. Detailed display under complex scenes.

From the results of the experiment and analysis, we believe that it is important to remove the irrational samples from the training set(Figure 9). The training set which is used at beginning, due to the automated cutting program, has a lot of images without vanishing point. This causes stagnancy of loss function after falling to a low level. By removing these interfering samples, the loss once again has a downward trend.

Fig 9. The detections of pictures without obvious vanishing point and detections of pictures with a obvious vanishing point.

4 CONCLUSION
In this paper, a detection method of vanishing point in complex and changeable unstructured circumstance has been developed based on CNN. In this paper, the training and testing datasets are collected and tagged manually. And then the structure of AlexNet is changed to suitable for regression, where the activate function is replaced by the Tanh, one fully connected layer are added, and the loss function is changed. The final detection results indicated the proposed method can locate the vanishing points and the accuracy is satisfied. And compare to classical method, this method is more effective on blurred pictures, dark scenes, rotating images and so on.

REFERENCES