Article

Development and Testing of a Scene-Compared Convolution Neural Network for Automatic Detection the Damage of Cameras in Monitoring System

Firstname Lastname [1,†,‡], Firstname Lastname [1,‡] and Firstname Lastname [2,*]

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; e-mail@e-mail.com
[*] Correspondence: e-mail@e-mail.com; Tel.: +x-xxx-xxx-xxxx
[†] Current address: Affiliation 3
[‡] These authors contributed equally to this work.

Academic Editor: name
Version November 11, 2017 submitted to Sensors

Abstract: Multiple cameras monitoring systems are used in many fields. When one or more of the cameras suffer damage, there would inevitably cause some missing of the key information. This paper proposes a novel method based deep convolutional neural networks (DCNN) to real-time detect the work situation of the cameras in a whole day. Different from previous DCNN which the net input only one image and output the classified label, show the camera is damaged or not, the proposed method input all of the cameras' images at once, those images are taken by the cameras in the same scene, then concatenate those images in their channel dimensionality, the process of the net can compare these images and readily find the abnormal cameras. The experiments and results demonstrate that the proposed method is not only able to attain a higher accuracy but convergence faster.

Keywords: convolutional neural networks; multiple cameras; damage detection; scene compare

1. Introduction

Surveillance cameras can be seen in everythere, schools, communities, markets and highway. All of these automated detetion systems, based on multi cameras, constitute an intelligent society. Some of the cameras are damaged will inevitably cause blind spots, as a result some scene information is missing and the function of the monitoring system is affected. Therefore, if the maintenance personnel be informed at the first time when the some of the cameras are damaged, the camera can be repaired at once and the loss will be minimized.

Currently, surveillance cameras can dignose the shelter, which make the image become blur because the lens is covered and is not in focus state (Figure 1). Through filter the image by laplacian operator, and then computer the average variance of the image matrix. If the result value less than a setted threshold meaning the image is blur and camera is convered by something. However, when lens is partially blocked, local damaged or the camera is subject to signal interference, the image shown as in Figure 2, will also cause monitoring blind spots, the algorithm needs to be detected this situation in real time. As the local blind spot's shape, color and location in the pictures are random and varied, it is difficult that use the features constructed by human to identify the damaged images. A well solution is that using deep convolution neural networks to classify these images to different category. DCNN trained by supervision method with substantial labeled images, can automaticly learn to extract image feature [1].

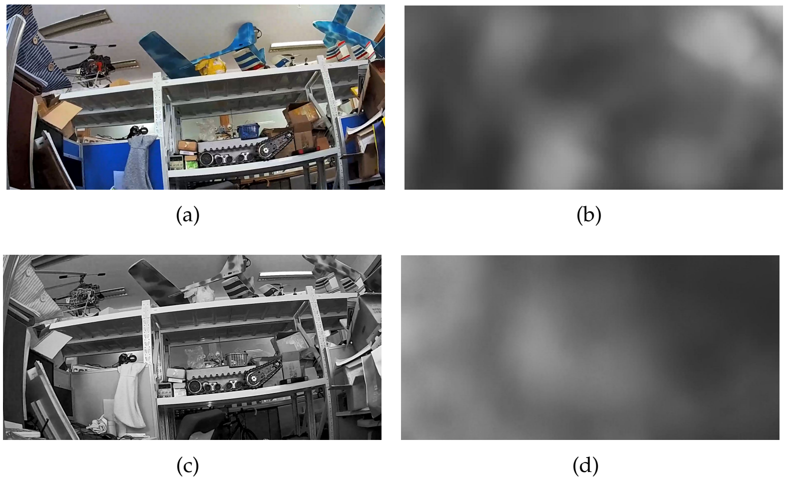
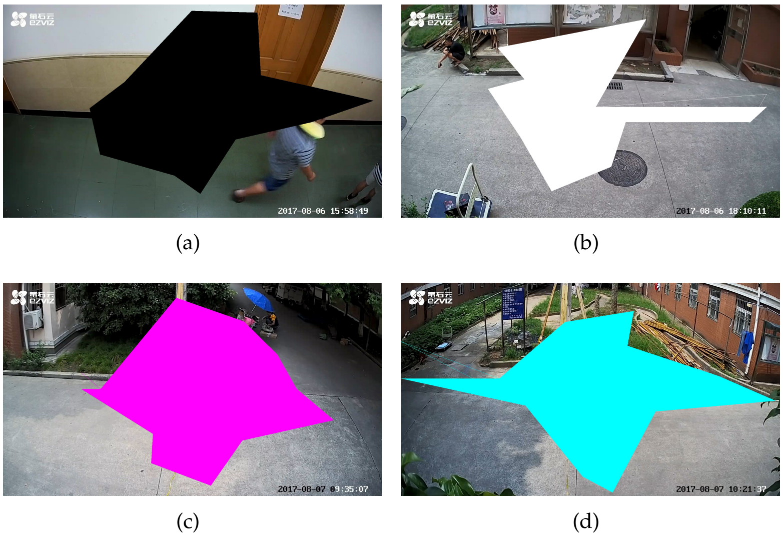

(a)
(b)
(c)
(d)

Figure 1. Examples of the normal images and lens covered images. (**a**) and (**b**) show a normal surveillance camera image taked in daytime and a image taked when the lens is covered by a piece of cloth. (**c**) and (**d**) show a normal surveillance camera image taked in night by Near-Infrared and a image taked when the lens is covered by a piece of cloth.



(a)
(b)
(c)
(d)

Figure 2. Examples of the blind spot images, which the spot's shape, color and location in images are diverse.

Since 2012 Alexnet won the Imagenet [2] title with a higher accuracy than the second [3], DCNN has been used in many areas, such as bearing fault diagnosis [4], traffic prediction [5], fruit detection [6] and et al. The common characteristics of above DCNN input is only one image, the network can extract the image feature and match it with the trained feature to identify the image. However, In multiple cameras monitoring system, for example in crossroads traffic monitoring system, cameras are in the same brightness, background and scene, when a camera is damaged, the image feature is significantly different from others. We can use this attribute to help us find out the damaged camera. So, we proposed a kind of convolution network, the main feature is that the network has multiple input, can compare these input cameras' images and real-time distinguish the damaged image. We call it Scene-Compared Convolution Neural Network. The proposed method input all of the cameras' images at once, those images are taken by the cameras in the same scene, then concatenate these images in their channel dimensionality, the process of the network can compare these images and readily find the abnormal cameras. The contributions of this paper are therefore:

- We proposed a Scene-Compared Convolution Neural Network that can readily discern the damaged cameras in multiple cameras monitoring system.
- The proposed network converges faster and has higher accuracy than other network architecture.
- We visualized the feature map of the network learned, proved the reason of the high-performance of the Scene-Compared Network architecture.

The rest of the paper is organized as follows: In Section 2, a two-step Scene-Compared Network that includes concatenating all cameras images to one in their channel dimensionality and some common CNN layers to process the "big image". In Section 3, experiments are conducted on the proposed Scene-Compared network, and are compared with the other prevailing methods. Finally, conclusions are drawn with future study directions in Section 4.

2. Methods

In the monitoring system, the camera images data distribution are highly similarity, so we construct the Scene-Compared Network. First, converting all images to one "big image". Then process the this "big image" with the common DCNN methods. Figure 3 illustrates the architecture of the proposed network. We will demonstrate the advantages of this architecture in the experiment section.

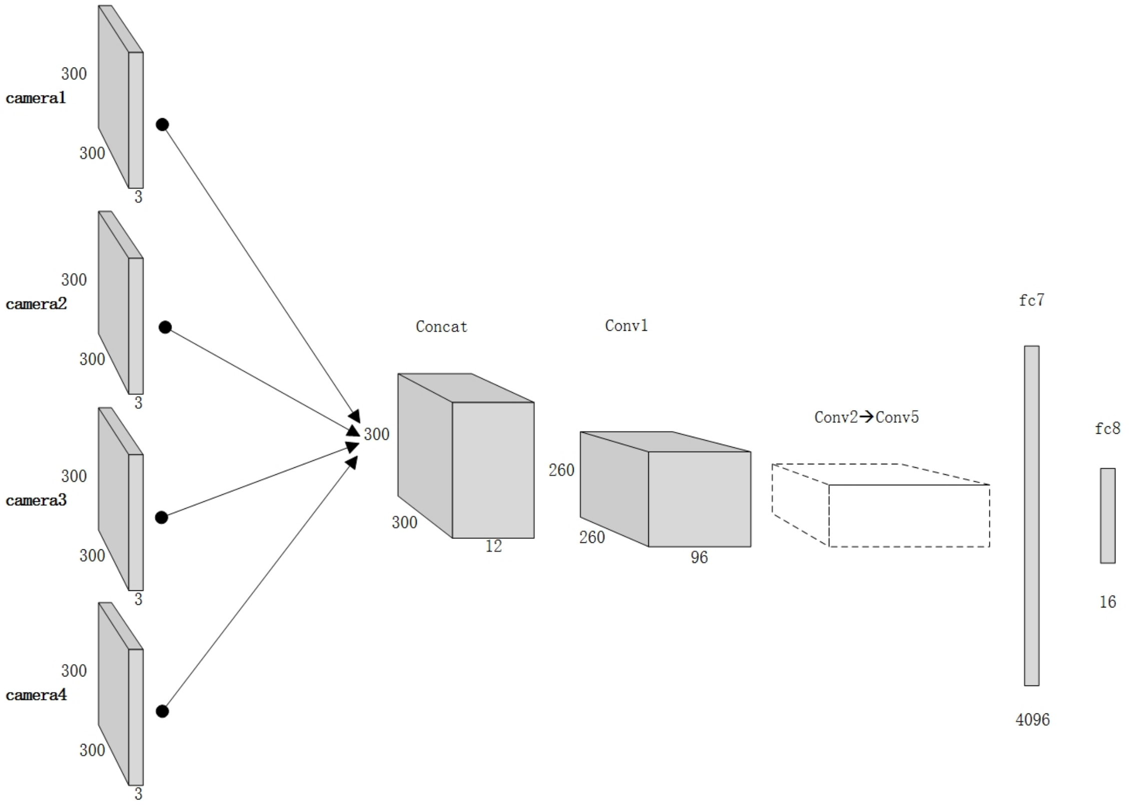


Figure 3. The Scene-Compared Network architecture.

2.1. Converting all images to one

As Figure 3 illustrated, The network input all of the cameras' images at once, then concatenate these images in their channel dimensionality. Let M_{ij}^{mn} represent the mth camera image, nth channel, ith row, jth column pixel, then we denote the "big image" matrix by:

$$\left[M_{i,j}^{11}, M_{i,j}^{12}, M_{i,j}^{13}, M_{i,j}^{21}, ..., M_{i,j}^{m1}, M_{i,j}^{m2}, M_{i,j}^{m3}\right] \tag{1}$$

In the training phase, label of the "big image" is decided by all labels of the source images. The merge rules are showed in Table 1.

Table 1. Labels merge rules.

label 1	label 2	label 3	label 4	result label
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
0	0	1	0	3
0	0	0	1	4
1	1	0	0	5
1	0	1	0	6
1	0	0	1	7
0	1	1	0	8
0	1	0	1	9
0	0	1	1	10
1	1	1	0	11
1	1	0	1	12
1	0	1	1	13
0	1	1	1	14
1	1	1	1	15

$$Big\ image\ label = rules\{labe1_1, label_2, ..., label_m\} \tag{2}$$

2.2. CNN Process

CNN Process include convolutional layer, active layer, pooling layer, and so on. It caculates the classification loss, throught back propagation and gradient descent to adjust the weights of the convolution kernels to make the kernels learn to extraction useful features. In the following part, we will describe these kinds of layers in more detail.

Convolutional layer is composed of a number of two-dimensional filters, each of the filters is a feature extractor. Suppose the input of the convolutional layer is $X^{N_{in} \times W_{in} \times H_{in}}$, meaning there are N_{in} channels, each of the channels' width is W_{in}, height is H_{in}. If the convolutional layer has n filters, the dimensional of thess filters are $w \times h$, then the output of the convolutional layer can be calculated as follows[7]:

$$C_j = \sum_{i=1}^{N_{in}} X_i * W_j + B_j \tag{3}$$

where C_j is the j-th channel of the convolutional layer; X_i represents the input data of the i-th channel; W_j is the weight of j-th filter; $*$ is an operator of convolution; B_j denotes the bias of the j-th channel. The number of the convolutional layer output channels are equal to the number of the filters. Suppose the convolution padding is p, striding is s, then the dimensional of the output of the convolutional layer ($N_{out} \times W_{out} \times H_{out}$) can be calculated as follows:

$$N_{out} = n \tag{4}$$

$$W_{out} = (W_{in} + 2 * p - w)/s + 1 \tag{5}$$

$$H_{out} = (H_{in} + 2 * p - h)/s + 1 \tag{6}$$

79 Activation layer as a nonlinear feature, can make the neural network approximate complex function, solve complex problems. It can be represent by the follows: 80

$$A_{out} = f(A_{in}) \tag{7}$$

81 For example, ReLu function[8]: $A(X) = max(0, X)$.

Pooling layer is a sub-sampling layer, the purpse of this layer is to improves the robustness of learned features through reducing the dimension of the input data. The common used pooling method is max pooling, it outputs only the maximum of each sub-sampling patch of the input channels to reduce the dimension of feature map. The output can be described as follows[7]:

$$P_{out} = \max_{Block \in P_{in}} \{Block\} \tag{8}$$

82 Suppose the *Block* size is $s \times s$, this function will pick one max value in each distinct s pooling block in the input data so that the output will become s times smaller along both spatial dimensions. 83

84 3. Experiments and Results

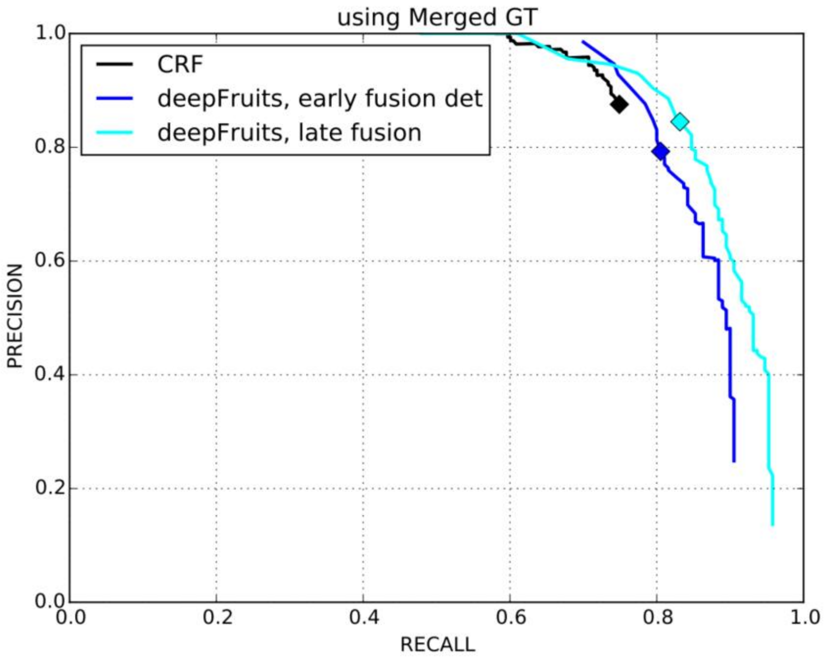
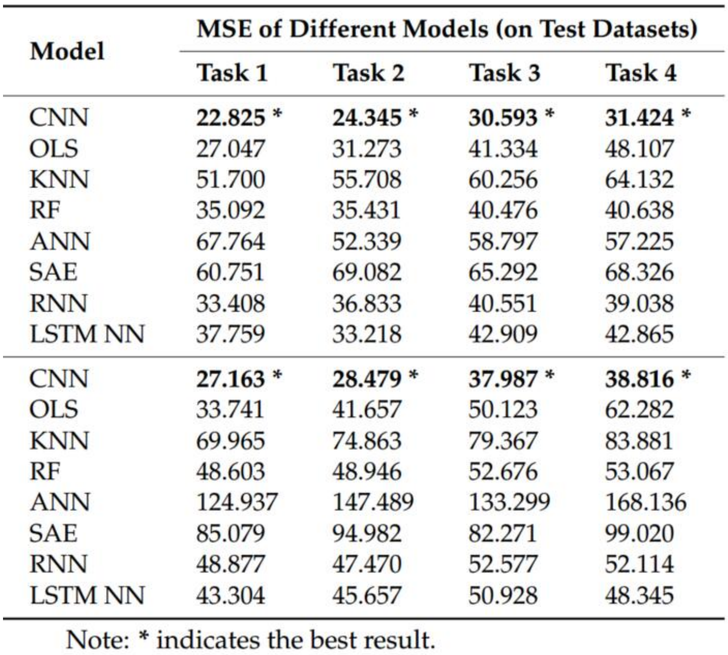


Figure 4. Accuracy of different methods.



Model	MSE of Different Models (on Test Datasets)			
	Task 1	**Task 2**	**Task 3**	**Task 4**
CNN	**22.825** *	**24.345** *	**30.593** *	**31.424** *
OLS	27.047	31.273	41.334	48.107
KNN	51.700	55.708	60.256	64.132
RF	35.092	35.431	40.476	40.638
ANN	67.764	52.339	58.797	57.225
SAE	60.751	69.082	65.292	68.326
RNN	33.408	36.833	40.551	39.038
LSTM NN	37.759	33.218	42.909	42.865
CNN	**27.163** *	**28.479** *	**37.987** *	**38.816** *
OLS	33.741	41.657	50.123	62.282
KNN	69.965	74.863	79.367	83.881
RF	48.603	48.946	52.676	53.067
ANN	124.937	147.489	133.299	168.136
SAE	85.079	94.982	82.271	99.020
RNN	48.877	47.470	52.577	52.114
LSTM NN	43.304	45.657	50.928	48.345

Note: * indicates the best result.

Figure 5. Accuracy on different size of the damage region in the images.

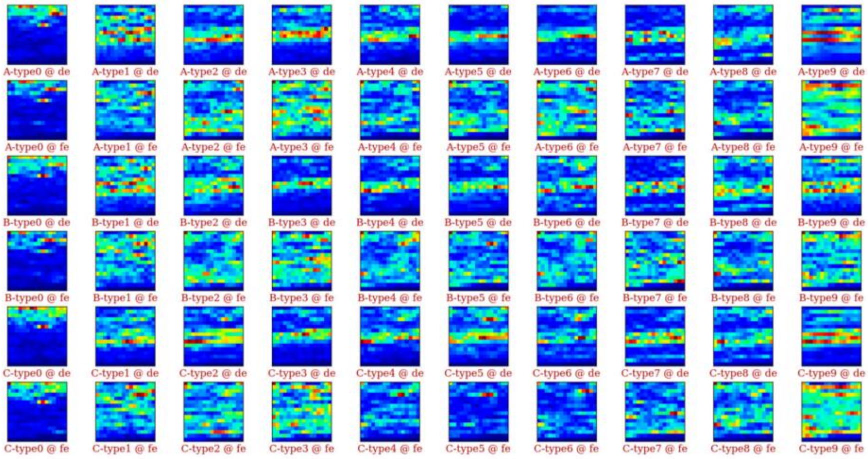


Figure 6. Visualize the feature map of the network.

.

4. Conclusions

The experiments and results demonstrate that the proposed method is not only able to attain a higher accuracy but convergence faster.

References

1. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. *ECCV* **2014**, *2313*, 818–833, DOI.
2. A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge 2010. www.image-net.org/challenges. 2010.

3. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS* **2012**.
4. Li, Shaobo, et al. An Ensemble Deep Convolutional Neural Network Model with Improved DS Evidence Fusion for Bearing Fault Diagnosis. *Sensors* **2017**, *1729*.
5. Yu, Haiyang, et al. Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. *arXiv preprint* **2017**, *1705.02699*.
6. Sa, Inkyu, et al. Deepfruits: A fruit detection system using deep neural networks. *Sensors* **2016**, *1222*.
7. Nebauer, C. Evaluation of convolutional neural networks for visual recognition. *IEEE Trans. Neural Netw.* **1998**, *9*, 685–696.
8. *******, C. Deep Sparse Rectifier Neural Networks. *******. *Neural Netw.* **1998**, *9*, 685–696.

Sample Availability: Samples of the compounds ...... are available from the authors.