Improve Neural Network Using Saliency

Yunong Wang[1,2(✉)], Nenghai Yu[1,2], Taifeng Wang[3], and Qing Wang[1,2]

[1] Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China
{wynd,wqing}@mail.ustc.edu.cn
[2] Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, Hefei 230027, China
ynh@ustc.edu.cn
[3] Microsoft Research, Beijing 100000, China
taifengw@microsoft.com

Abstract. In traditional neural networks for image classification, every input image pixel is treated the same way. However real human visual system tends pay more attention to what they really focus on. This paper proposed a novel saliency-based network architecture for image classification named Sal-Mask Connection. After learning raw feature maps from input images using a convolutional connection, we use the saliency data as a mask for the raw feature maps. By doing an element-by-element multiplication with the saliency data on the raw feature maps, corresponding enhanced feature maps are generated, which helps the network to filter information and to ignore noise. By this means we may simulate the real human vision system more appropriately and gain a better performance. In this paper, we prove this new architecture upon two common image classification benchmark networks, and we verify them on the STL-10 datasets. Experimental results show that this method outperforms the traditional CNNs.

Keywords: Sal-Mask connection · Machine learning · Convolutional neural network · Saliency · Image classification

1 Introduction

Convolutional neural network (CNN) has been very popular and proved extremely effective in handling the problems of image recognition, object detection and image classification. In image classification, it has become the state-of-art. Li Wan [1] got an error rate of 0.21 % and Ciresan [2] got 0.23 % on the MNIST dataset, beating all other methods such as SVM, K-Nearest Neighbors, Boosted Stumps and those Neural Nets, none of which use convolutional connections.

The structure of CNNs is under fast and varied progress, but since LeNet-5 [3], it has become a typically standard structure for CNNs - stacked convolutional connections with one or more full connections at the end of the network. Within the net, convolutional connections are optionally followed by contrast normalization and

Y.-J. Zhang (Ed.): ICIG 2015, Part II, LNCS 9218, pp. 417–429, 2015.
DOI: 10.1007/978-3-319-21963-9_38

pooling connections. Many kinds of variants of this basic structure are proposed, and get the best results [4,5] so far on MNIST, CIFAR and ImageNet classification challenge. Network in Network (NIN) [6] and GoogleNet [7] reflect the trend that nets are becoming bigger and deeper while becoming more and more better. Using stacked convolutional connections as the main frame, researchers have discovered different ways to improve the performance. Alex came up with the local response normalization in NIPS 2012 [4]. Hinton proposed dropout [8] and then dropconnect [1] generalized after it. Also a novel way to train activation functions adaptively named Maxout [9] is proposed by Goodfellow. Despite all those variants of CNNs, they are all in fact the same - to stack and combine convolutional connections, pooling connections, full connections and others to form a net. A network like that can't simulate human visual system (HVS) well enough because the input image pixels of the CNNs are treated evenly. By that, it means that each pixel is treated as importantly as one another, which is handled quite differently in real HVS.

In real HVS, we tend to pay more attention to what we really focus on when we look. It is generally agreed that the HVS uses a combination of image driven data and prior models in its processing [11]. What HVS used is called Visual Saliency. The ability of the HVS to detect visual saliency is extraordinarily fast and reliable [10]. Saliency is a broad term that refers to the idea that certain parts of a scene are pre-attentively distinctive and create some form of immediate significant visual arousal within the early stages of the HVS [11]. It has been a central problem in many computer vision tasks, and helps to solve matching or correspondence problems such as object recognition, classification or tracking. Though the neural network is attempting to mimic how the HVS might be processing the images, computational modeling of this basic intelligent behavior remains a challenge.

To take further advantage of saliency information, we proposed a new connection named Sal-Mask Connection in this paper, which uses saliency data as an element-by-element mask for raw feature maps learned by convolutional connection from input images. The Sal-Mask Connection helps the neural networks work better. We call the network with a Sal-Mask Connection in it as a Sal-Mask Network.

In Sect. 2, we will explain the detail of a Sal-Mask Connection. Section 3 illustrates the structure of two traditional CNNs we used as benchmarks and the structure of the Sal-Mask Networks we used to verify the improvement. In Sect. 4, several experimental results will be presented. Section 5 will include a short conclusion and a precise introduction of some future work.

2 Sal-Mask Connection and Sal-Mask Network

2.1 Backgrounds of Saliency

It is observed that primates have a remarkable ability to interpret complex scenes in real time, despite the limited speed of the neuronal hardware available for such tasks [12]. It is generally acknowledged that the HVS uses a combination of image driven data and prior models in its processing, which means that intermediate

and advanced visual processes select a subset of the available sensory information before further processing, most likely to reduce the complexity of scene analysis. This selection appears to be implemented in the form of a spatially circumscribed region of the visual field, the so-called focus of attention, which means that particular locations in the scene are selected based on their behavioral relevance or on local image cues [13]. Saliency is the metadata that describes the cues, it plays an important role in HVS, helps human to understand what they see.

Applying that procedure of HVS to a neural network system - the first convolutional connection is like the first routine of HVS, and the rest part of the network is like the further processing of HVS. As human will always select particular locations on the first routine using saliency information, it is natural to ameliorate the neural network in the same way. It is obvious that a better saliency will lead to a better performance. But it is hard to say which saliency is best for a neural network, as basically neural network is a black box lacking details inside. While there is no explicit rule to follow, it is a trade off to take a trial-way to judge the saliency algorithms. However, for lack of computation resources and time, we only tried out a few kinds of saliency data and we will chose two of them as testimony. One type of saliency we used is from Dani et al. [14], in the following paper, we will call it Alpha Saliency for short. The other is from Mingming Chen et al. [15], we will call it RC Saliency for short, as Chen called it so in [15]. Figure 1 has some examples.

Fig. 1. Saliencies. (a) is original image, (b) is Alpha saliency and (c) is RC saliency.

2.2 Structure of Sal-Mask Network

Before the discussion begins, as the concepts of layer and connection are often comprehended differently and confusing in different theses, we define the data of neurons within the same layer as a Layer in this paper, and the data of weights and computation along with it as a Connection, as shown in Fig. 2.

As described before, typically a CNN network starts with a convolutional connection. Based on that, we divide a convolution neural network into two parts. One includes the input layer, the first convolutional connection and the second layer - the feature maps learned by the first convolutional connection, which we named as raw feature maps. And the other one which includes the rest part of the network, we call it a Basic Neural Network (Basic NN).

To simulate the procedure of HVS as has been noted, the structure of so-called Sal-Mask Network is shown in Fig. 3. In Fig. 3, layer 1 is original image

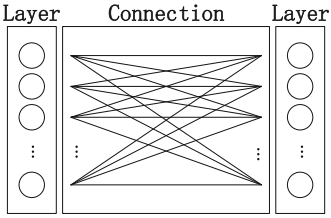

Fig. 2. Definition of layer and connection

data, layer 2 is saliency data which is generated using some pre-defined saliency algorithms. Layer 3 contains raw feature maps learned from layer 1 using a convolutional connection. Combine raw feature maps from layer 3 and saliency data from layer 2, we get the enhanced feature maps, as is layer 4. And the connection before layer 4 is is what we called Sal-Mask connection, which helps enhance the raw feature maps. Then the enhanced feature maps are input into the rest of the net. This is how a Sal-Mask Network works. More details about the Sal-Mask Connection and Sal-Mask Network will be introduced in Sect. 2.3.

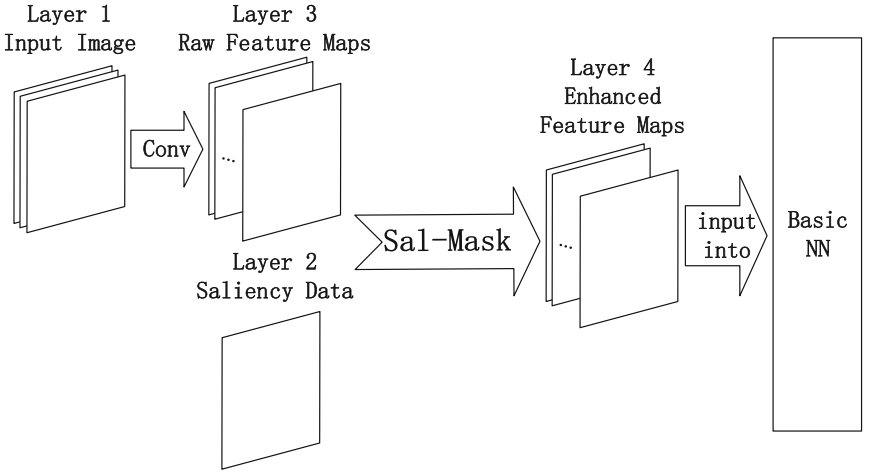

Fig. 3. Structure of Sal-mask network

2.3 Formulas and Deduction

As in layer 1, we use x_i to indicate the i-th pixel of the input image. Here we treat the two-dimension plus multi-channel image as a one-dimension array, as it is equivalent on math. Thus $i \in \{1, N\}$, where $N = imageWidth \cdot imageHeight \cdot imageChannel$.

Layer 2 is saliency data, which are obtained using some pre-defined algorithms. In this paper, it is either RC saliency or Alpha Saliency. Also, we

treat the two-dimension saliency data as a one-direction array. Hence s_i indicates the i-th pixel of the saliency data. The width and height of saliency data is the same as that of the input image data, hence $i \in \{1, M\}$, where $M = imageWidth \cdot imageHeight$.

Layer 3 is raw feature maps learned from layer 1 using convolutional connection. y_{jk} is the value of the j-th neuron of the k-th raw feature map. As in this paper, we'll acquiesce to do a padding operation to keep the feature maps the same size as the input image. Here, each feature map, which is two-dimension, is treated as a one-direction array, while there are K raw feature maps, K depends on the detail of the convolutional connection. Then we expand the formula of the convolutional connection to a common format, using w_{ijk} to indicate the shared weight between x_i and y_{jk}:

$$z_{jk} = \sum w_{ijk} \cdot x_i + b_{jk}. \tag{1}$$

$$y_{jk} = f(z_{jk}). \tag{2}$$

f is the activation function of the convolutional connection, and z_{jk} is the value of neuron before activation function. b_{jk} is bias.

The most essential part is the Sal-Mask Connection before layer 4. It takes both layer 2 and layer 3 as inputs. For each raw feature map in layer 3, it makes an element-by-element multiplication with the saliency in layer 2, and generates a corresponding enhanced feature map in layer 4. Therefore, the enhanced feature maps are calculated as follows:

$$ez_{jk} = y_{jk} \cdot s_j. \tag{3}$$

$$ey_{jk} = h(ez_{jk}). \tag{4}$$

ey_{jk} is value the j-th pixel of the k-th enhanced feature map. ez_{jk} is the value before activation function and h is the activation function used.

As Eq. (3) described the feedforward of Sal-Mask Connection, we can deduce the the backpropagation formula using the definition of the backpropagation algorithm. δ_{jk} is the error backforwarded to layer 3 and layer 2 needs no feedback as it is an input layer. By expanding δ_{jk} using chain rule, we get:

$$\delta_{jk} = -\frac{\partial E}{\partial z_{jk}} = -\frac{\partial E}{\partial ez_{jk}} \cdot \frac{\partial ez_{jk}}{\partial y_{jk}} \cdot \frac{\partial y_{jk}}{\partial z_{jk}}. \tag{5}$$

According to the definition of Eq. (3), we know that

$$\frac{\partial ez_{jk}}{\partial y_{jk}} = s_j. \tag{6}$$

And with Eq. (2) we get

$$\frac{\partial y_{jk}}{\partial z_{jk}} = f'(z_{jk}). \tag{7}$$

Combining all the above, we get the formula for the backpropagation like this:

$$\delta_{jk} = e\delta_{jk} \cdot sy_j \cdot f'(z_{jk}). \tag{8}$$

$e\delta_{jk}$ is the error of layer 4, which is passed backward from the rest of the network, δ_{jk} is the error of layer 3.

It can be observed from the formulas before, that on both feedforward and backpropagation processes, the higher the value of saliency (s_j) is, the more it contributes to the output and the more effective feedback we will get. However, in original saliency data as shown in Fig. 1, there are too many pixels with value 0. Thus the corresponding neurons of raw feature maps will make no contribution to the feedforward result and gain no feedback during backpropagation. So we use the average of saliency data and a pure white image as the mask for Sal-Mask Connection to guarantee that the mask will not cause too many loss of the details of input images.

3 Network Structures

3.1 Benchmark Networks and Sal-Mask Networks Improved

As mentioned before, stacked convolutional layers have become a typical standard structure for a CNN since LeNet-5 [3]. We chose two typical examples out of them and do the experiments based on these two networks: AlexNet [4] and NIN [6]. However, due to the limit of GPU we used, we changed the structure slightly and used a simplified version with minor parameters. Besides, we skip the data augmentation which will take too much time on computation. So it may not perform as well as the state-of-art technology, still the comparison results are clear enough to prove the Sal-Mask Connection and Sal-Mask Network effective. The network structure of AlexNet and the corresponding Sal-Mask Network based on it are shown in Fig. 4. The structure of NIN and Sal-Mask Network based on it are shown in Fig. 5.

3.2 Connections and Parameters

In Figs. 4 and 5, a square frame stands for a Layer. As we use our networks on image classification problems, the size of a Layer is in format of $imageWidth \cdot imageHeight \cdot imageChannels$, except for the last frame, which is the output layer and the number stands for how many classes the dataset has. Texts nearby arrows between two layers stand for Connections, with the illustration of connection type, serial number and parameters.

Conv is short for convolutional connection. It has three parameters: kernel size $m \cdot n$, kernel number K and stride s. In Figs. 4 and 5, the parameters are in format of $m \cdot n \cdot K$ and the stride is set to 1. Before every convolutional connection, we do a padding to keep the corner pixels of the input images at the center of the kernels. That makes the feature maps the same size of image before the convolutional connection. Relu is short for Rectified Linear Units. It is used as activation function instead of sigmoid and linear. It is used by Alex [4] too. In the network, every convolutional connection and full connection used Relu.

Pool is short for pooling connection. It has three parameters: pooling size k, pooling stride s and pooling function - whether max-pooling(short as max) or

Fig. 4. Structure of Alex and Sal-Mask Network derided from it. (a) is AlexNet, (b) is the Sal-Mask Network derided from AlexNet.

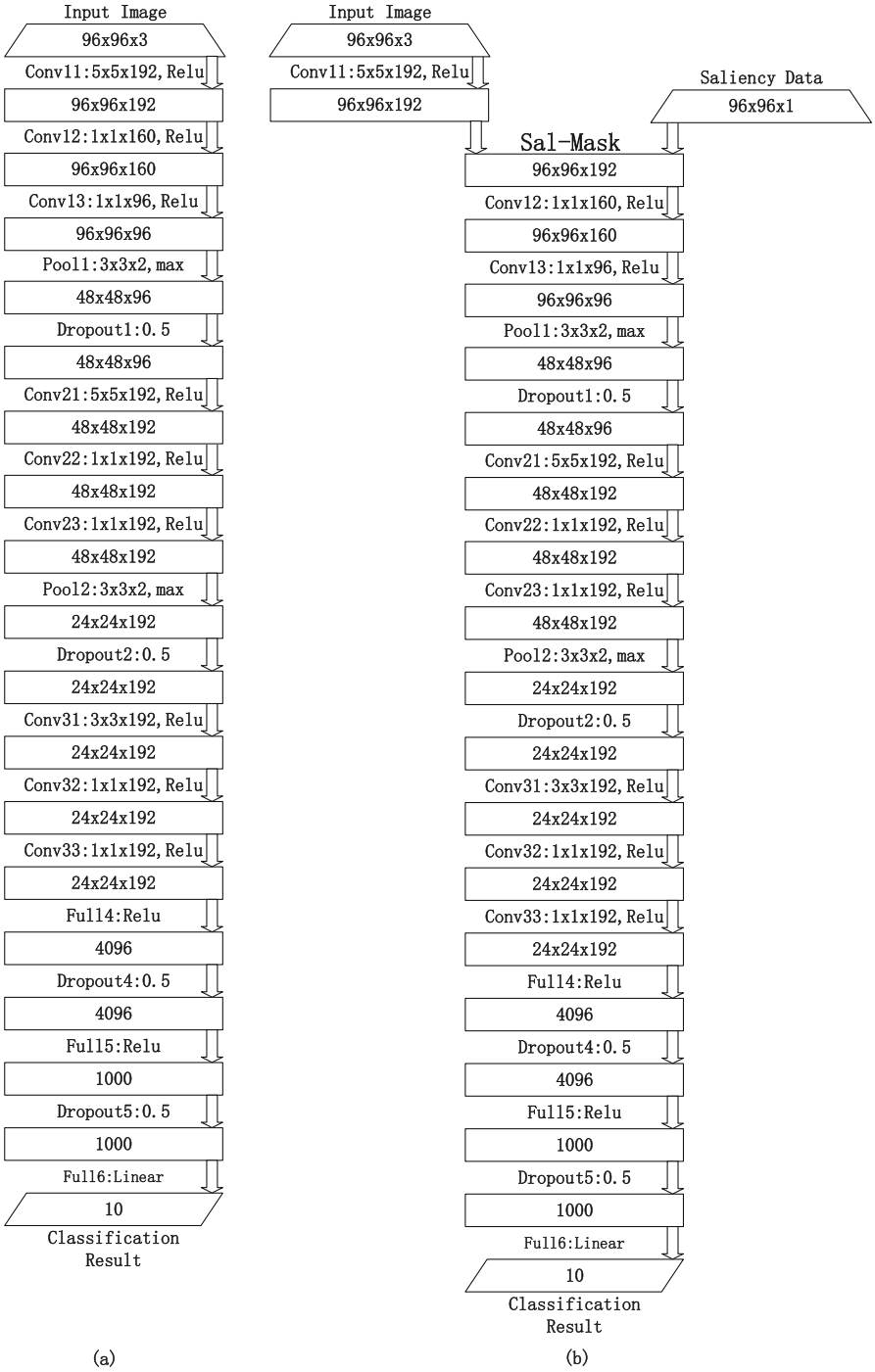

Fig. 5. Structure of NIN and Sal-Mask Network deride from it. (a) is NIN, (b) is the Sal-Mask Network derided from NIN.

average pooling(short as ave). In Fig. 4, the parameters are in format of $k \cdot k \cdot s$, and pooling function is described. Basically, all pooling connections we used have the parameter of $k = 3$, $s = 2$ and are max-poolings. They are overlapping pooling connections, as same as AlexNet [4].

LRNorm is short for local response normalization which is also proposed by Alex [4]. Denoted by $a^i_{x,y}$ the activity of a neuron computed by applying kernel i at position (x, y), the response-normalized activity $b^i_{x,y}$ is given by the expression

$$b^i_{x,y} = a^i_{x,y}/(k + \alpha \cdot \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^i_{x,y})^2)^\beta. \tag{9}$$

where the sum runs over n adjacent kernel maps at the same spatial position, and N is the total number of kernels in the layer. Local response normalization connection has four parameters : k, n, α and β. In our experiments, we set k as 1, n as 3, α as 0.00005 and β as 0.75.

Full is short for full connection, which depends on the size of both input layer and output layer. And Dropout is for dropout connection, which drops neurons of input layer radomly by ratio r. It is proposed by Hinton [8]. In our experiments, we set r as 0.5 for every dropout connection.

4 Experiments

4.1 Code and Dataset

Caffe [16] is a deep learning framework developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. We implemented our own code based on the design of Caffe, and we implemented the code for Sal-Mask Connection by our own.

As the memory and computation ability of GPU we used is limited, we are unable to afford a run on the ImageNet dataset. Therefore, we have to pick a minor dataset instead. However, saliency algorithms so far are not good enough for images that are too small, so the dataset we use cannot be too small. Taking these two points into consideration, we used STL-10 dataset [17]. It is an image recognition dataset inspired by the CIFAR-10 dataset but with some modifications. The size of images in it is 96×96 pixels, which is ok for saliency algorithms. The labeled data in it includes 5000 training images and 8000 test images, which makes the dataset not too big to run a network on it. As the name suggests, there are 10 classes of images in STL-10.

4.2 Improvement on AlexNet

Using structure described in Fig. 4, we use fixed learning rate 0.001 at the beginning and turn it down to 1/10 every 40 epochs. We do the test every epoch and get a benchmark on STL-10 using AlexNet, which is shown by solidline in Fig. 6. Then we put Sal-Mask Connection on AlexNet to improve it into a Sal-Mask Network. We get the result shown in Fig. 6, and it proved Sal-Mask Connection

do make the performance better. The accuracy for Sal-Mask Network using RC Saliency is drawn by short dashed line, and the accuracy for Sal-Mask Network using Alpha Saliency is drawn by long dashed line in Fig. 6. It is clear to see that Sal-Mask Network do improve the AlexNet.

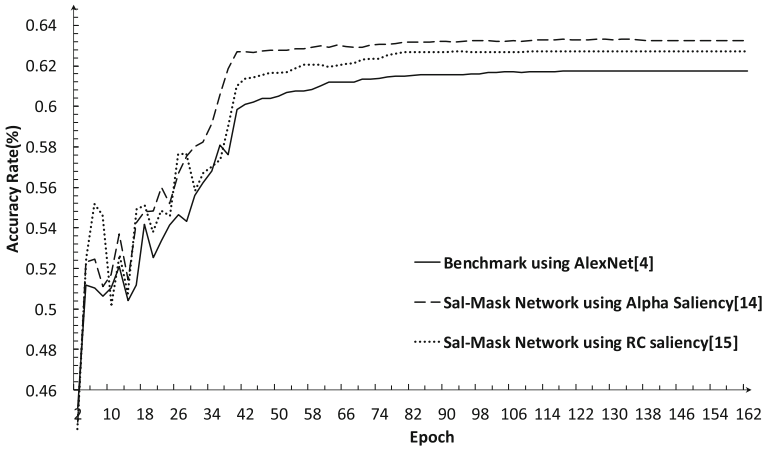

Fig. 6. Results for three different nerworks

Table 1 shows the best accuracy for each experiment, and the better results are highlighted in bold. Using RC Saliency as mask, the Sal-Mask Network based on AlexNet is 0.98 % better than original AlexNet. And the Sal-Mask Network using Alpha Saliency is 1.5 % better than original AlexNet.

Table 1. Improvement on AlexNet

Network	Accuracy(%)
Benchmark using AlexNet	61.74
Sal-Mask Network using RC saliency	**62.72**
Sal-Mask Network using Alpha saliency	**63.24**

4.3 Improvement on NIN

Using the structure described in Fig. 5, we use fixed learning rate 0.001 at the beginning and turn it down to 1/10 every 40 epochs. Again we do the test every epoch and get a benchmark on STL-10 using NIN, which is shown by solidline in Fig. 7. Then we put Sal-Mask Connection on AlexNet to improve it into a Sal-Mask Network. The accuracy for Sal-Mask Network using RC Saliency is drawn by the short dashed line, and the accuracy for Sal-Mask Network using Alpha Saliency is drawn by the long dashed line in Fig. 7. The Sal-Mask Connection also works well on the base of NIN.

Table 2 shows the best accuracy for each experiment, and the better results are highlighted in bold. The accuracy increased 2.10 % using RC Saliency and 0.70 % using Alpha Saliency.

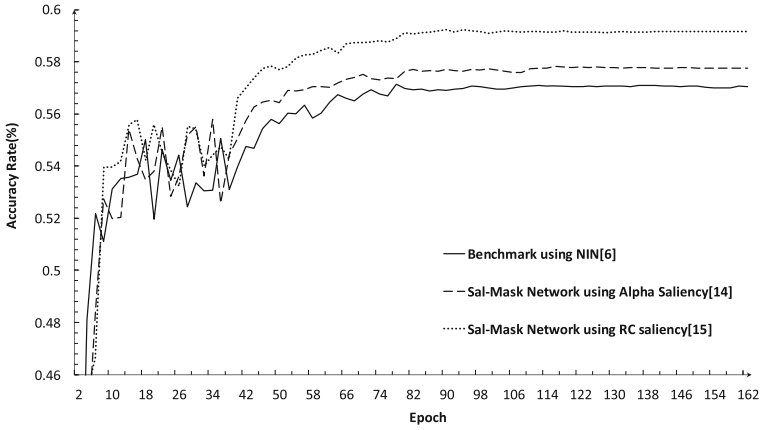

Fig. 7. Results for three different nerworks

Table 2. Improvement on NIN

Network	Accuracy(%)
Benchmark using NIN	57.13
Sal-Mask Network using RC saliency	**59.23**
Sal-Mask Network using Alpha saliency	**57.83**

4.4 Discussion

By results shown before, it is clear that with networks with Sal-Mask Connection work better than the benchmark networks. With different saliency data as mask, the performance may improve differently, varying from 0.70 % to 2.10 % as the results shows. With these two groups of experiments as proof, it is convinciable that Sal-Mask Connection will work on different CNNs.

5 Conclusion

In this paper, we proposed a novel connection called Sal-Mask Connection, which is build upon the idea that the HVS would select a subest of the input visual information. To simulate that procedure, we use saliency data as a element-by-element mask for raw feature maps learned from input images by the first convolutional connection of the network. The new connection helps the network to filter information and ignore noise. It is theoretically correct when designing the structure, according to saliency theory for HVS. And it is also proved correct by solid experiment results. In this paper, we use the proposed connecion on two common image classification benchmark networks, and experimental results show that this method is superior to the traditional CNNs. With the help of the experiemnts, Sal-Mask Connection is proved useful as it can work on neural

networks using a convolutional connection at the very beginning. CNNs have proven the best for recognition, classification and more other tasks, while Sal-Mask Connection can help CNNs to work better.

However, there is still more research to take up. First of all, the saliencies used as mask are chosen manually and it is decided with a trial-error strategy, so it is urgent to find out one that works the best. Additionally, it should be determined whether there is any rule for the choice of appropriate saliency? Second, since human would learn to recognize objects for years during childhood, it seems necessary and fair for neural networks to use some prior knowledge for help. However, is there some way that is more effective to take advantage of these prior knowledge? Yet, is there any other information that can be taken into account apart from saliency data? Sal-Mask is only a small step for development of neural networks, a lot more questions remain to be answered.

Acknowledgments. This work is supported by the National Science Foundation of China (No.61371192, No.61103134).

References

1. Wan, L., Zeiler, M., Zhang, S., et al.: Regularization of neural networks using dropconnect. In: International Conference on Machine Learning, pp. 1058–1066 (2013)
2. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition, pp. 3642–3649 (2012)
3. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems, pp. 1097–1105 (2012)
5. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833 (2014)
6. Lin, M., Chen, Q., Yan, S.: Network in network (2013). arXiv preprint arXiv:1312.4400
7. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A., et al.: Going deeper with convolutions (2014). arXiv preprint arXiv:1409.4842
8. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors (2012). arXiv preprint arXiv:1207.0580
9. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks (2013). arXiv preprint arXiv:1302.4389
10. Hou, X., Zhang, L.: Saliency detection: a spectral residual approach. In: Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
11. Kadir, T., Brady, M.: Saliency, scale and image description. Int. J. Comput. Vis. **5**(2), 83–105 (2001)
12. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)

13. Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. Vis. Res. **40**(10), 1489–1506 (2000)
14. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 228–242 (2008)
15. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: Computer Vision and Pattern Recognition (CVPR), pp. 409–416 (2011)
16. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Darrell, T., et al.: Caffe: convolutional architecture for fast feature embedding. In: ACM International Conference on Multimedia, pp. 675–678 (2014)
17. Coates, A., Ng, A.Y., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: International Conference on Artificial Intelligence and Statistics, pp. 215–223 (2011)