Ada-Sal Network: emulate the Human Visual System

Yunong Wang^{a,b,*}, Nenghai Yu^{a,b}, Taifeng Wang^c

^a*Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China*

^b*Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences*

^c*Microsoft Research, Beijing 100000, China*

Abstract

Convolutional neural networks (CNN) have become state-of-the-art for image classification. Inspired by the physiological mechanism of saliency in real human visual system (HVS), we had previously proposed the Sal-Mask Connection. As HVS tends to select specified region of the visual field depending on saliency to interpret complex scenes, we use saliency data as an element-by-element mask on feature maps learned from convolutional connections. The effectiveness of the Sal-Mask Connection had been verified in our previous work. However, as the performance of Sal-Mask Connection was influenced by the saliency data used, and current saliency algorithms are not designed to work for image classification, it is urgent and essential that we obtain a more suitable saliency. In this paper, therefore, we propose Ada-Sal Network to learn saliency adaptively while feature maps are being trained at the same time. Experiments on Cifar-10 and STL-10 datasets are done with three various networks. In each test, we compare the performance of benchmark network, Sal-Mask Networks using two different saliencies, with that of the Ada-Sal Network. The results indicate that Ada-Sal Network outperforms not only traditional networks but also Sal-Mask Network using non-adaptive saliency. Visualization of the networks exhibits the saliency learned adaptively is able to combine merits from input saliencies and seems to work better for most cases.

Keywords: Ada-Sal Network, Convolutional Neural Network, Human Visual System, Image Classification, Machine Learning, Saliency, Sal-Mask Connection.

1. INTRODUCTION

With the popularity of Neural networks, the convolutional neural networks stand out as a new star on image recognition, object detection and image classification and so on. Wan et al. [1] got an error rate of 0.21% and Ciresan et al. [2] got 0.23% on the MNIST dataset, outperformed all other methods such as SVM, K-Nearest Neighbors,

*Corresponding author

Email addresses: wynd@mail.ustc.edu.cn (Yunong Wang), ynh@ustc.edu.cn (Nenghai Yu), taifengw@microsoft.com (Taifeng Wang)

Boosted Stumps and those Neural Nets without convolutional connections. He et al. [3] claimed an error rate of 4.94% surpassing the human-level performance(5.1% [4]).

The structure of CNNs is under fast and varied progress, but since LeNet-5 [5], it has become a typical 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 pooling connections. Variants of this basic structure are proposed, and get the best results ([6][7]) so far on MNIST, CIFAR and ImageNet classification challenge. Network in Network(NIN) [8] and GoogleNet [9] explicitly suggests that nets are becoming bigger and deeper with their better performance. Using stacked convolutional connections as the main frame, researchers have discovered different ways to improve the performance. Krizhevsky et al. proposed the local response normalization in NIPS 2012 [6]. Hinton et al. proposed dropout [10] and then generalized dropconnect [1] after it. A novel way to train activation functions adaptively named Maxout [11] was proposed by Goodfellow et al. Despite all those variants of CNNs, they share one thing in common: to stack and combine convolutional connections, pooling connections, full connections and others to form a net.

The development of neural networks benefits a great deal from the researches into Human Visual System. At first, neural network was proposed to emulate human brain and the structure of neurons. Then Convolutional connection was inspired by receptive field. Nowadays, there is a certain mechanism called visual attention mechanism in Human Visual System that can be employed to help improve the neural networks. Human developed this mechanism owing to the limitation of human brain: we human beings pay attention to only part of the visual scene and focus on it since we cannot process unlimited visual information. It is generally agreed that the HVS uses a combination of image driven data and prior models in its processing [13], which is called Visual Saliency. The HVSs capacity to detect visual saliency is extraordinarily fast and reliable [12]. 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 [13]. It has always been a center in many computer vision tasks, and is of great importance when it comes to matching or correspondence problems, for instance, 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.

As we try to reform the neural networks by emulating the idea of saliency, we propose a new structure named Sal-Mask Connection [14] which takes two inputs. A saliency map is inputted into the Sal-Mask Connection and is used as a mask. The other input of the Sal-Mask Connection is raw feature maps learned by a convolutional connection. The Sal-Mask Connection performs an element-by-element multiplication between the saliency map and the raw feature maps. Thanks to that, we get enhanced feature maps with better features to proceed in the other part of the neural networks. We call a network using Sal-Mask Connection a Sal-Network. This is what we did previously in Wang et al. [14]. In this paper, we extend our theory of Sal-Mask Connection and describe it as a mechanism of region selection and resource allocation. As traditional saliency detection algorithms we used do not work well enough, especially when the input images are too small, we put forward a new structure based on

the Sal-Network. We use a convolutional connection to combine two input saliencies together and learn a new mask adaptively. The new Sal-Network equipped with the ability to learn saliency adaptively is called Ada-Sal Network. Details would be introduced in the following chapters. We also do more experiments and analyze the process by visualizing the feature maps learned in the neural networks.

In Section 2, we explain the details of Ada-Sal Network, including design idea and deduction of formulas. Section 3 deals with three traditional CNNs we employed as benchmarks and Sal-Network derived from each benchmark network. Experimental results are presented and analyzed in Section 4. A brief conclusion as well as our future plan is drawn in Section 5.

2. Sal-Mask Connection

Before the discussion begins, as the concepts of layer and connection are often comprehended differently and confusing in different theses, we call the neurons within the same layer as a Layer in this paper, and the weights and computation along with it as a Connection, as shown in Figure 1.

Figure 1: Definition of Layer and Connection

2.1. The Computer Visual Model of CNNs

Given the fact that neural network is proposed to emulate the human brain, and that convolutional neural network is inspired by the idea of receptive field in the HVS, obviously, the development of the computer visual model concerning neural networks is inspired by the discoveries from researches on the real human visual system. So the structure of CNNs can be reflected in the structure of HVS.

As described before, typically a CNN network starts with a convolutional connection, so we divide a convolution neural network into two parts. The former part includes the input layer, the first convolutional connection and the second layer. The second layer is raw feature maps learned by the first convolutional connection. The latter part includes the rest of the network. The whole network can be interpreted as extracting raw features from input images, then training a classifier based on those raw features. So we call the former part a feature-extractor and the latter part a Basic NN, as depicted in Figure 2.

Figure 2: Structure of Sal-Mask Network

2.2. Background of Saliency

With a better understanding of HVS, 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 [15]. 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 [16]. Saliency, the metadata used to describes the cues, plays a vital role in HVS, helping human to understand what they see.

In conclusion, it is essential and beneficial to modify the computer visual model of CNNs according to the idea of saliency as the understanding of HVS varies. As saliency can be explained as the ability to choose more important visual areas and apply more resources in comprehending these scenes, we propose Sal-Mask Connection and Ada-Sal to mimic HVS more appropriately.

2.3. New Saliency-based Model: Ada-Sal Network

To emulate the region-select procedure of HVS, we added a Sal-Mask Connection between the two parts of neural networks, as shown in Figure 3, where layer 1a is original image data, layer 1b is some saliency maps which are generated by using some pre-defined saliency algorithms. Layer 2a contains raw feature maps learned from layer 1a using a convolutional connection. Layer 2b is the new saliency learned from layer 1b and is used as the mask for the Sal-Mask Connection. Layer 2a and layer 2b are input into the Sal-Mask Connection. After an element-by-element multiplication, we get enhanced feature maps in layer 3. This structure replaced the original feature extractor. As a result, we now can get enhanced feature maps instead of raw feature maps. Then the enhanced feature maps, as input into the Basic NN, can be classified. The element-by-element multiplication operation of Sal-Mask Connection is designed

to emulate the region-select function. As after Sal-Mask Connection, the features in the area with high saliency are reserved while the features at the area with low saliency are depressed.

Figure 3: Structure of Ada-Sal Network

In this paper, we used two kinds of saliencies as input for our algorithm. One type of saliency we used is from Levin et al. [17], which is called Alpha Saliency for short in the following sections. The other is from Cheng et al. [18], we call it RC Saliency for short. Figure 4 shows some examples.

2.4. Deduction of BP Algorithm for Sal-Mask Connection

As in layer 1a, we use x_i to indicate the i -th pixel of the input image. Here we treat the two-dimensional plus multi-channel image as a one-dimensional array, as it is equivalent on math. Thus $i \in \{1, N\}$, where $N = W \cdot H \cdot C$, where W is the width of the image, H is the height of the image and C is the number of channels of the image.

Layer 2a is a mask for the Sal-Mask Connection. The mask is a two-dimensional matrix with the same size of raw feature maps in layer 2a. Also, we treat the two-

Figure 4: Saliencies. (a) is original image, (b) is Alpha Saliency and (c) is RC Saliency.

dimensional data as a one-dimensional array. Hence s_i indicates the i -th pixel of the mask, and $i \in \{1, M\}$, where $M = W \cdot H$.

Layer 2a are raw feature maps learned from layer 1a using convolutional connection. y_{jk} is the value of the j -th neuron of the k -th raw feature map. As in this paper, a padding operation is intended to keep the feature maps the same size as the input image. Here, each feature map, which is two-dimensional, is treated as a one-dimensional array, while there are K raw feature maps, K depending 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} :

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

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

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

$$E_{jk}^z = y_{jk} \cdot s_j. \quad (3)$$

$$E_{jk}^y = h(E_{jk}^z). \quad (4)$$

E_{jk}^y is the value of the j -th pixel of the k -th enhanced feature map. E_{jk}^z 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 back-propagation formula using the definition of the back-propagation algorithm. δ_{jk} is the error back forwarded to layer 2a and δ_j^s is the error back forwarded to layer 2b. δ_j^s is calculated as feedback for the adaptive learning of the mask. By expanding δ_{jk} and δ_j^s using chain rule, we get:

$$\delta_{jk} = -\frac{\partial E}{\partial z_{jk}} = -\frac{\partial E}{\partial E_{jk}^z} \cdot \frac{\partial E_{jk}^z}{\partial y_{jk}} \cdot \frac{\partial y_{jk}}{\partial z_{jk}}. \quad (5)$$

$$\delta_j^s = -\frac{\partial E}{\partial s_j} = -\sum_k \frac{\partial E}{\partial E_{jk}^z} \cdot \frac{\partial E_{jk}^z}{\partial s_j}. \quad (6)$$

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

$$\frac{\partial E_{jk}^z}{\partial y_{jk}} = s_j. \quad (7)$$

$$\frac{\partial E_{jk}^z}{\partial s_j} = y_{jk}. \quad (8)$$

With Eq. (2) we get

$$\frac{\partial y_{jk}}{\partial z_{jk}} = f'(z_{jk}). \quad (9)$$

Taking all mentioned above into account, we get the formula for the back-propagation like this:

$$\delta_{jk} = \delta_{jk}^e \cdot s y_j \cdot f'(z_{jk}). \quad (10)$$

$$\delta_j^s = \sum_k \delta_{jk}^e \cdot y_{jk}. \quad (11)$$

δ_{jk}^e is the error of layer 3, which is passed backward from the rest of the network, δ_{jk} is the error of layer 2a and δ_j^s is the error of layer 2b.

It now can be observed from the previous formulas that in 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 get.

3. Network Structures

3.1. Benchmark Networks

As mentioned before, stacked convolutional layers have become a typical standard structure for a CNN since LeNet-5 [5]. We chose three typical examples and do the experiments based on these three networks: Mod-LeNet [5](we modified the structure of LeNet-5 a little: we removed last two full connections, based on the advice of Caffe), AlexNet [6] and NIN [8]. However, due to the limit of GPU we used(Nvidia GTX970), we changed the structure slightly and reduced the number of kernels of convolutional connections. Besides, we skipped the data augmentation which will take too much time on computation. So it may not perform as well as the state-of-the-art technology, yet the comparison results are clear enough to prove the efficiency of our method. Then we derive two Sal-Mask Networks using different saliency. The structures of 3 benchmark networks are shown in Figure 5 and the Sal-Network derived is illustrated in Figure 6.

3.2. Connections and Parameters

In Figure 5 and Figure 6, a square frame stands for a Layer. When we use our networks on image classification problems, the size of a Layer is in format of $W \cdot H \cdot C$ where W is the width, H is the height and C is the number of channels, 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 Figure 5 and Figure 6, 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, making 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

Figure 5: Structure of 3 benchmark networks on STL-10: (a) is the Mod-LeNet, (b) is the AlexNet and (c) is the NIN.

sigmoid and linear. It is used by Krizhevsky et al. [6] 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 average pooling(short as ave). In Figure 5, 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-pooling. They are overlapping pooling connections, the same as AlexNet [6].

LRNorm is short for local response normalization which is also in Krizhevsky et al. [6]. Denoted by $a_{x,y}^i$ the activity of a neuron computed by applying kernel i at position (x,y) , the response-normalized activity $b_{x,y}^i$ is given by the expression

$$b_{x,y}^i = a_{x,y}^i / (k + \alpha \cdot \sum_{j=\max(0, i-n/2)}^{\min(N-1, i+n/2)} (a_{x,y}^j)^2)^\beta. \quad (12)$$

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. Dropout is for dropout connection, which drops neurons of input layer randomly by ratio r . It is proposed by Hinton et al. [10]. In our experiments, we set r as 0.5 for every dropout connection.

Figure 6: Structure of Sal Network on STL-10. It replaces the structure in the solid-line frame in Figure 5 with the structure in the dash-line frame correspondingly and gets 3 Sal-Network for each benchmark network.

3.3. Sal-Network and Ada-Sal Network Derived

We split the traditional neural network as feature extractor and Basic NN. The part in the solid-line frames in Figure 5 are feature extractors. In this paper Sal-Network and Ada-Sal Network are both tested. If we add a Sal-Mask Connection into the networks and replace the original feature extractors with structure in the dash-line frame, then we get our Sal-Network derived from each benchmark network, as shown in Figure 6. If we replace the original feature extractors with structure in the solid-line frame in Figure 7, then we get our Ada-Sal Network derived from each benchmark network.

Figure 7: Structure of Ada-Sal Network on STL-10. It replaces the structure in the solid-line frame in Figure 5 with the structure in the solid-line frame correspondingly and gets 3 Ada-Sal Network for each benchmark network.

4. Experiments

In this section, we will introduce the experiments we have done and present the results. We chose 2 datasets and 3 kinds of network as benchmark. With each benchmark, we did three comparison experiments: two are Sal-Mask Network using two different saliencies as mask, while another is the Ada-Sal Network we proposed in this paper. Details are introduced as follows.

4.1. Code and Datasets

Caffe [19] 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 on our own.

We chose two datasets for our experiments - CIFAR-10 [20] and STL-10 [21]. The CIFAR-10 dataset consists of 60000 32×32 pixels color images in 10 classes, including 50000 training images and 10000 test images. The STL-10 dataset is inspired by the CIFAR-10 dataset but with some modifications. The images are at the size of 96×96 pixels. The labeled data in it includes 5000 training images and 8000 test images for 10 classes.

We demonstrate our classification performances with three networks: Mod-LeNet, AlexNet and NIN on two databases: CIFAR-10 and STL-10. We use these three traditional networks for benchmarks. As contrast experiment, we also do more experiments based on the idea from our previous work where we used the saliency as mask directly. The results are shown and analyzed below.

4.2. Pre-processing on Saliency

As discussed before, in 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 get. However, in original saliency data as shown in Figure 4, there are too many pixels with value 0. Thus the corresponding neurons of raw feature maps make no contribution to the feedforward result and gain no feedback during

backpropagation. So when the saliency data is used in the Sal-Network directly, some pre-processing is done on it. First we normalize the saliency data to the scale of 0 to 1. Then we add value 1 on every pixel and divide the sum by 2. We avoided the value 0 by doing so. But when saliencies are used in the Ada-Sal Network, there is no need for pre-processing on the saliencies. Instead, we take the Alpha saliency, the RC saliency and an image with value 1 on every pixel to learn a new saliency adaptively.

Table 1: Results on CIFAR-10

Network-Accuracy (%)	Mod-LeNet	AlexNet	NIN
Benchmark	84.6 $\pm$ 0.1	83.5 $\pm$ 0.2	83.3 $\pm$ 0.3
Sal-Mask Network using RC saliency	83.7 $\pm$ 0.2	82.2 $\pm$ 0.3	81.8 $\pm$ 0.2
Sal-Mask Network using Alpha saliency	84.7 $\pm$ 0.2	83.1 $\pm$ 0.2	83.3 $\pm$ 0.3
Ada-Sal Network	85.6$\pm$0.2	84.7$\pm$0.2	85.5$\pm$0.3

Table 2: Results on STL-10

Network-Accuracy (%)	Mod-LeNet	AlexNet	NIN
Benchmark	70.0 $\pm$ 0.3	61.7 $\pm$ 0.2	61.8 $\pm$ 0.2
Sal-Mask Network using RC saliency	71.7 $\pm$ 0.2	63.4 $\pm$ 0.2	62.2 $\pm$ 0.3
Sal-Mask Network using Alpha saliency	71.4 $\pm$ 0.2	62.7 $\pm$ 0.2	61.2 $\pm$ 0.3
Ada-Sal Network	72.8$\pm$0.3	64.2$\pm$0.3	62.8$\pm$0.2

4.3. Results on CIFAR-10

For the experiments on CIFAR-10, the learning rate starts with a certain value for each network and reduces to 1/10 every 40 epochs. We run 162 epochs for the network to converge.

Figure 8-10 show the accuracy curves of all the 3×4 experiments we introduced in Section 3. Here we take the AlexNet in Figure 8 as example, for benchmark Mod-LeNet network, the accuracy converges and reaches the maximum value of 84.59% after 150 epoches. For Sal-Mask Network using adaptively learned mask, the accuracy reaches the maximum value of 85.52% at the 92th epoch and finally converges to 85.45%. However, the Sal-Mask Networks using saliency data directly as the mask performed no better: the one using Alpha Saliency directly as mask only reached 84.63% and converged to 84.38%, while the one using RC saliency directly as mask only reached 83.48%. The reason for the failure of Sal-Mask network using saliency directly as mask is that images in CIFAR-10 are too small, and that the saliency algorithms are not good enough to get appropriate saliency for the input images. However,

by learning the mask adaptively from two kinds of not-good-enough saliency, with the help of feedback from the neural network, we learn mask that is good enough to improve the performance.

Table 1 shows the best accuracy for each experiment, and the better results are highlighted in bold.

Figure 8: Performances on CIFAR-10 using four different networks with Mod-LeNet as benchmark

Figure 9: Performances on CIFAR-10 using four different networks with AlexNet as benchmark

4.4. Results on STL-10

For the experiments on STL-10, the learning rate starts with a certain value for each network and reduces to 1/10 every 40 epochs. We also run for 162 epochs for the network to converge. As STL-10 only has 5000 images for training, the problem of overfitting is quite severe. With the number of parameters of the networks increasing from Mod-LeNet to AlexNet to NIN, the performance keeps decreasing. Figure 11-13 show the accuracy curves and we skip the interpretation of the curves here.

Figure 10: Performances on CIFAR-10 using four different networks with NIN as benchmark

Figure 11: Performances on STL-10 using four different networks with Mod-LeNet as benchmark

Figure 12: Performances on STL-10 using four different networks with AlexNet as benchmark

Figure 13: Performances on STL-10 using four different networks with NIN as benchmark

Table 2 shows the performances for each experiment, and the better results are highlighted in bold.

4.5. Discussion

From results shown above, it is clear to see that: 1) the Ada-Sal Network improves the classification accuracy of benchmarks indeed, and 2) the Ada-Sal Network works better than those Sal-Mask Networks using saliency directly.

4.5.1. Effect of Sal-Mask Connection on Features

The reason for 1 is that Sal-Mask Connection is built upon the idea of simulating the area-selection procedure of HVS. Processed by Sal-Mask Connection, features from more relevant area are enhanced and features from irrelevant area that may disturb the classification are suppressed. In that way, the Sal-Mask Connection helps to train better parameters and focus on relevant information and filter noises.

Fig. 14 shows how the Ada-Sal Network can get better features trained. As shown in Fig. 14(a), it is a picture of a horse. This picture is incorrectly classified in the benchmark network we trained, whereas it is correctly recognized in the Ada-Sal Network. There are three main objects in the picture: the horse, the tree behind and the grassland on the background. The saliency learned from Ada-Sal shows that the area of main object—the horse—is obtained and marked brighter than backgrounds. (We will discuss this adaptively learned saliency latter, but here it is fair to say that this saliency is workable.) According to feature maps in Fig. 14(b), we can see three main kinds of features, one for each main object: features for the horse are at the main part of the picture, features for the tree are mainly on the top-left area and features for the grassland are mainly on the right side of the picture. Obviously, the features for the tree and the grassland would be a distraction for the classifier, as tree may also appear in the pictures of birds and grassland may appear in the pictures of dogs and deer.

As we designed, the Sal-Mask Connection works in two aspects: to enhance features for main object of the image, and to suppress features for irrelevant backgrounds. Fig. 14 proves that Sal-Mask Connection works exactly as we wanted:

- Compare the feature maps in Fig. 14(b) with those in Fig. 14(c), we can see that in the Ada-Sal Network, features for the tree and the grassland are suppressed, while features for the horse are amplified.
- Compare the feature maps in Fig. 14(d) with those in Fig. 14(c), we visualize the effect of Sal-Mask Connection. The 1st, 4th and 5th ones show the ability of amplifying features for the horse. The 2nd, 3rd, 4th and 5th feature maps show the ability of depressing the features for the tree and the grassland.
- Compare the feature maps in Fig. 14(b) with those in Fig. 14(d), we find that not only will the forward operation of Sal-Mask Connection affect the features learned, but the feedback from Sal-Mask Connection affects the training as well. After 160 epochs of training, the distinction between the raw feature maps gained in the Ada-Sal Network and the feature maps from the benchmark network is big enough to see.

All these feature maps prove that the existence of Sal-Mask Connection in the neural network helps to get better features out of the input image.

Besides, it is interesting that even though the benchmark network and the Ada-Sal Network are trained independently, the features they get are quite similar with each other, with only minor difference mainly on the features for the tree and the grassland. However, with similar features, this image is classified correctly in the Ada-Sal Network rather than in the benchmark network. It can be explained as the features of the tree and the grassland distract the classifier, as mentioned previously.

Figure 14: (a) A picture of horse in STL-10 dataset, and the saliency learned adaptively from Ada-Sal Network based on Mod-LeNet, (b) 5 typical raw feature maps learned by the first convolutional connection of Mod-LeNet. (c) 5 similar enhanced feature maps learned by Ada-Sal Network. (d) corresponding raw feature maps for feature maps in (c).

4.5.2. Saliency from Ada-Sal Network

According to the cases provided in last section, we can draw the conclusion that Sal-Mask Connection is able to improve the neural network. However, the result somehow depends on the saliency learned adaptively. If the adaptively learned saliency works well, the Ada-Sal Network will work well too. So it is important that whether the Ada-Sal Network is able to get saliency and train features good enough at the same time.

Fig. 15 shows five images that are incorrectly classified in benchmark Mod-LeNet Network while they are classified correctly in Ada-Sal Network. As the illustrations show, the Ada-Sal Network is able to learn a better saliency. A complex linear adder is trained to combine both two saliencies. The adder usually combines the advantage of either kind of saliency and gets better saliency. It only fails when both saliencies are quite bad. Only in such cases, the adaptively learned saliency is unsatisfying.

As we don't have the ground truth data for the STL-10 dataset, we are unable to compare the Alpha Saliency, the RC saliency with the adaptively learned saliency quantitatively. But we can analyze them qualitatively anyhow. Take Fig. 15 as examples, the Alpha Saliency and the RC Saliency may miss some part of the main object while marking, each of which misses different parts. Combine both saliencies and we can get a saliency that makes sense for describing the whole image.

Fig. 16(a) are five randomly picked images from CIFAR-10. As we can see, RC saliency algorithm works well for image 1 and 3, however, it totally gets wrong for image 2, 4 and 5. Whereas, Alpha saliency algorithm works well for image 2, 4 and 5, but it performs poorly on image 1 and 3. According to result in Fig. 9, the Sal-Mask Network performs worse using RC or Alpha Saliency as a result of the incapability of both saliencies to identify the important parts of images. However, the saliency learned adaptively works well on most cases and improves the performance of the benchmark network, as shown in Table. 1.

5. Conclusion

In this paper, we used Sal-Mask Connection on traditional CNNs to emulate HVS more appropriately. In order to get a good saliency to describe the importance of pixels in the input images, we proposed the approach to learn mask adaptively using a convolutional connection on several available saliencies. The new structure is named Ada-Sal Network.

It is theoretically correct when designing the structure, according to saliency theory for HVS. It also proved correct by sound experiment results. We tested our idea upon two common image classification datasets with three different benchmark networks, and experimental results showed that Ada-Sal Network is superior to the traditional CNNs. The saliency learned adaptively proved better than saliencies calculated by those two pre-defined algorithms.

In short, we conclude that Ada-Sal Network is able to get a measure for the importance of pixels in the input images which are better than using the saliency data directly. The Sal-Mask Connection works the way it is designed to: it helps to amplify features from area belonging to the semantic object of the image. Meanwhile,

Figure 15: (a) are 5 win-case images from STL-10, (b) are RC saliencies for these images, (c) are Alpha saliencies for these images and (d) are masks learned adaptively in the modified network based on Mod-LeNet.

Figure 16: (a) are 5 images from CIFAR-10, (b) are RC saliencies for these images, (c) are Alpha saliencies for these images and (d) are masks learned adaptively in the modified network based on Mod-LeNet.

features from irrelevant areas are depressed. It is demonstrated that Ada-Sal Network is an effective method to help filter irrelevant information and focus on main object of the input images. As CNNs have proven effective and still show potential on the tasks of recognition, classification and more other tasks so far, Sal-Mask Connection and Ada-Sal Network have proven useful to help CNNs to work better.

In the future, we expect to use several different saliencies as input for the Ada-Sal Network and see how far it can go. What's more, we are considering calculating saliency using neural network itself. That means we do not need to calculate saliencies as extra input for training anymore. If we figure out how, we can get better saliency and better features at the same time directly from an Ada-Sal Network.

6. 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. International Conference on Machine Learning. 1058–1066(2013)
- [2] Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3642–3649(2012)
- [3] He, K., Zhang, X., Ren, S. and Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision. 1026-1034(2015)
- [4] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), pp.211-252(2015)
- [5] 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 computation. 1(4), 541–551(1989)
- [6] Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems. 1097–1105(2012)
- [7] Zeiler, M. D., Fergus, R.: Visualizing and understanding convolutional networks. European Conference on Computer Vision. 818–833(2014)
- [8] Lin, M., Chen, Q., Yan S.: Network in network. arXiv preprint arXiv:1312.4400(2013)

- [9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–9(2015)
- [10] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R.: Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*(2012)
- [11] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. *International Conference on Machine Learning*. (2013)
- [12] Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. *Computer Vision and Pattern Recognition (CVPR)*. 1–8(2007)
- [13] Kadir, T., Brady, M.: Saliency, scale and image description. *International Journal of Computer Vision*. 5(2), 83–105(2001)
- [14] Wang, Y., Yu, N., Wang, T., et al.: Improve Neural Network Using Saliency. *International Conference on Image and Graphics*, 417–429(2015)
- [15] Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 20(11), 1254–1259(1998)
- [16] Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*. 40(10), 1489–1506(2000)
- [17] Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 30(2), 228–242(2008)
- [18] Cheng, M., Zhang, G., Mitra, N., Huang, X., Hu, S. M.: Global contrast based salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 409–416(2011)
- [19] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Darrell, T., et. al.: Caffe: Convolutional architecture for fast feature embedding. *ACM International Conference on Multimedia*. 675–678(2014)
- [20] Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 1(4), 7(2009)
- [21] Coates, A., Ng, A. Y., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. *International Conference on Artificial Intelligence and Statistics*. 215–223(2011)
- [22] Simonyan, K., Vedaldi, A. and Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations*. (2014)

- [23] Pan, H. and Jiang, H.: A Deep Learning Based Fast Image Saliency Detection Algorithm. arXiv preprint arXiv:1602.00577(2016)