Predicting Eye Fixations using Convolutional Neural Networks

Nian Liu[1], Junwei Han[1*], Dingwen Zhang[1], Shifeng Wen[1] and Tianming Liu[2]
[1]Northwestern Polytechnical University, P.R. China
[2]University of Georgia, USA
{liunian228,junweihan2010,zhangdingwen2006yyy, wenshifeng90}@gmail.com
tliu@cs.uga.edu

Abstract

It is believed that eye movements in free-viewing of natural scenes are directed by both bottom-up visual saliency and top-down visual factors. In this paper, we propose a novel computational framework to simultaneously learn these two types of visual features from raw image data using a multiresolution convolutional neural network (Mr-CNN) for predicting eye fixations. The Mr-CNN is directly trained from image regions centered on fixation and non-fixation locations over multiple resolutions, using raw image pixels as inputs and eye fixation attributes as labels. Diverse top-down visual features can be learned in higher layers. Meanwhile bottom-up visual saliency can also be inferred via combining information over multiple resolutions. Finally, optimal integration of bottom-up and top-down cues can be learned in the last logistic regression layer to predict eye fixations. The proposed approach achieves state-of-the-art results over four publically available benchmark datasets, demonstrating the superiority of our work.

1. Introduction

When viewing visual scenes, human visual system has the ability to selectively locate eye fixations on some informative contents. In computer science field, researchers normally develop computational visual saliency models to quantitatively predict human eye attended locations using computer vision techniques. In recent years, a large number of computational models [1-7] and applications [8-10] have been proposed.

Inspired by the biological evidence that locations distinctive from their surroundings are more likely to attract our attention, most traditional approaches typically cope with the problem of saliency modeling by three steps in sequence: early feature extraction, feature contrast inference, and contrast integration. For early feature extraction, Itti *et al.* [1] proposed three low-level features including intensity, color, and orientation. Judd *et al.* [2] considered more early features, for example, subbands of the steerable pyramid, 3D color histograms and color probabilities based features. These works rely on hand-crafted features. To acquire powerful hand-crafted features, sufficient and proper domain-specific knowledge is generally required. Nevertheless, a thorough understanding of human visual attention mechanisms has not been achieved yet. Meanwhile, the hand-crafted features may be not universally appropriate for different types of images. Although some machine learning methods have been involved in some models, e.g. ICA [11] and sparse coding [4, 12, 13], it's still very hard for these models to mine high-level information and latent patterns of complex images due to the limited representational capability of their shallow architectures.

In saliency models, contrast computation over early features is another key procedure. Itti *et al.* [1] designed the "center-surround difference" operator across multiple scales to calculate contrast. Later on, a lot of works followed Itti's idea to compute contrast from different views via using a variety of mathematical tools, for example, using information theories [12], frequency spectrum [14-17], sparse coding [13, 39] or autoencoder [40]. From these previous works, we can see that most of them resort to human-designed mechanisms to calculate contrast, which would be insufficient to handle large-scale data with complex distributions.

The last step for saliency modeling is to integrate various contrast features to yield saliency maps. Itti *et al.* [1] linearly fused three contrast maps using fixed weights. Zhao and Koch [18] learned the optimal weights associated with various contrasts using a least square technique upon a set of eye tracking data. Judd *et al.* [2] learned a linear SVM to fuse bottom-up features. Similarly, Borji [3] explored the linear regression model and AdaBoost classifier for optimized feature fusion.

Although most previous works mainly concentrate on contrast-based bottom-up saliency, it is believed that at early stage of free viewing, eye movements are mainly directed by bottom-up visual saliency and later on, by high-level factors (e.g., objects [19, 20], actions [21], and events) [22, 23]. Thus it is inevitable to combine bottom-up saliency information and top-down factors to build a

* Corresponding author.

superior model for predicting eye fixations. Some works have pursued this direction. For instance, Cerf *et al.* [24] combined face detection with low-level saliency. Judd *et al.* [2] and Borji [3] combined bottom-up features with more top-down factors, including humans, faces, cars, texts, and animals. Although these methods achieve better performance than traditional models relying on visual saliency alone, there is still much room for improvement because only a small number of hand-tuned factors are used in these models.

All of the issues discussed above motivate us to design a new unified learning model to enhance the hand-crafted bottom-up saliency features and top-down factors for eye fixation prediction. To this end, this paper proposes a novel computational model based on a multiresolution convolutional neural network (Mr-CNN) which simultaneously learns early features, bottom-up saliency, top-down factors, and their integration from raw image data. To be specific, as shown in Figure 1 and Figure 2, we train a Mr-CNN directly from image regions centered on fixation and non-fixation locations over multiple resolutions, using raw image pixels as inputs and eye fixation attributes as labels. Benefitting from its hierarchical architecture and the purely supervised training manner, our model can learn saliency-related features with hierarchically increasing complexity in convolutional layers, instead of resorting to various hand-crafted features. These features learned with hierarchical depth can represent original image regions efficiently and discriminatively. In higher layers, the proposed Mr-CNN can learn diverse high-level top-down visual features due to its deep architecture. Meanwhile, it can also learn bottom-up saliency via combining information over multiple resolutions. Considering local image regions with the same center location but with fine-to-coarse resolutions (see the three image regions of the traffic sign in Figure 1), finer image regions are actually the central parts of coarser ones. When the deep features of both the center (the finer image region) and the context (the coarser image region) are inputted to a neural network simultaneously, the difference between them may be learned under the supervision of labels, which makes the proposed Mr-CNN have the capability to learn the bottom-up saliency, contrary to using various human-designed mechanisms in traditional models. Finally, the last logistic regression layer learns to integrate bottom-up saliency with top-down cues to predict eye fixations.

We notice that some researchers have applied deep learning algorithms to model visual saliency lately. Shen *et al.* [25] used a 3-layer convolutional sparse coding model to learn high-level concepts from fixated image regions in an unsupervised way. Then, a linear SVM was utilized to detect saliency from those learned concepts. Similarly, Vig *et al.* [26] learned a linear SVM over hierarchical optimal features which are optimized via a bio-inspired hierarchical models (hierarchical neuromorphic networks) in their Ensemble of Deep Networks (eDN) model. These two methods mainly focus on learning deep features while ignoring the importance of bottom-up visual saliency. Lin *et al.* [27] used a set of adaptive convolutional low-level filters learned by k-means algorithm to produce low-level and mid-level features. Then, the center-surround difference was performed over the learned features to compute local contrast. In this model, top-down factors were not taken into account. In contrast to these previous works, our proposed work builds a unified framework to learn both the bottom-up saliency and top-down factors simultaneously.

The contributions of this paper can be summarized as follows. (1) By using a Mr-CNN, we implement the learning of early features, bottom-up saliency, top-down factors, and their integration from image data itself simultaneously. The yielded method is automated and does not depend on hand-tuned features or calculation mechanisms. (2) The proposed method is evaluated on four widely used eye-tracking benchmark datasets and achieves better results compared to 11 state-of-the-art models. (3) We visualize the learned hierarchical features from the Mr-CNN. It demonstrates that the proposed Mr-CNN can learn both low-level features related to bottom-up saliency and high-level top-down factors to improve eye fixation prediction. Furthermore, the learned features can also uncover novel insights for the psychophysics of fixation selection and the intrinsic biological mechanism, which we wish can offer novel inspiration to explore the human vision system.

The rest of this paper is organized as follows. Section 2 describes the proposed model using the Mr-CNN. Section 3 reports the quantitative and qualitative experimental results on four benchmarks. Finally, we draw conclusions in Section 4.

2. Proposed model

In this section, we elaborate the approach we propose. As illustrated in Figure 1 and Figure 2, the model architecture is mainly based on a Mr-CNN. We first briefly review CNN, and then we depict the proposed Mr-CNN in details and show how to use it to predict eye fixations.

2.1. A brief review of CNN

A convolutional neural network (CNN) [28] is usually composed of alternate convolutional and max-pooling layers (denoted as C layers and P layers) to extract hierarchical features to represent the original inputs, subsequently with several fully connected layers (denoted by FC layers) followed to do classification.

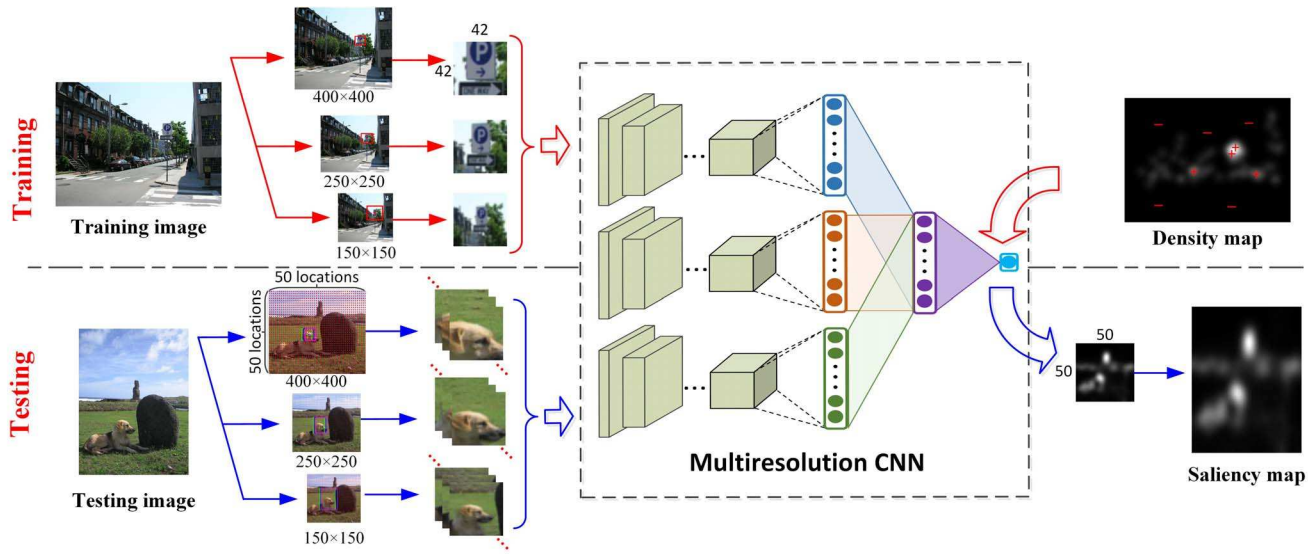

Figure 1: Diagram of our Mr-CNN based model. First, the given image is rescaled to three scales, i.e. 150×150, 250×250 and 400×400, then 42×42 sized image regions with the same center locations are extracted from the rescaled image duplicates as inputs to the Mr-CNN. We extract fixation and non-fixation image regions to train the Mr-CNN. When testing, we just evenly sample 50×50 locations per image to estimate their saliency values to reduce computation cost. The obtained down-sampled saliency map is rescaled to the original size to achieve the final saliency map.

Considering a CNN with L layers, we denote the output state of the l-th layer as $\mathbf{H}_l$, where $l \in \{1,...,L\}$, additionally using $\mathbf{H}_0$ to denote the input data. There are two parts of trainable parameters in each layer, i.e. the weight matrix $\mathbf{W}_l$ that connect the l-th layer and its previous layer with state $\mathbf{H}_{l-1}$, and the bias vector $\mathbf{b}_l$.

The input data is usually connected to a C layer. For a C layer, a 2D convolution operation is performed first with convolutional kernels $\mathbf{W}_l$. Then the bias term $\mathbf{b}_l$ is added to the resultant feature maps, in which a pointwise non-linear activation operation $Actv$ is typically performed subsequently. Finally a max-pooling layer is usually followed to select the dominant features over non-overlapping square windows per feature map. The whole process can be formulated as:

$$\mathbf{H}_l = pool(Actv(\mathbf{H}_{l-1} * \mathbf{W}_l + \mathbf{b}_l)), \quad (1)$$

where $*$ denotes the convolution operation and $pool$ denotes the max-pooling operation.

Several C layers and P layers can be stacked one by one to form the hierarchical feature extraction architecture. Then, the resultant features are further combined into 1D feature vectors by several FC layers. A FC layer first processes its inputs with linear transformation by weight $\mathbf{W}_l$ and bias $\mathbf{b}_l$, then the pointwise non-linear activation is followed:

$$\mathbf{H}_l = Actv(\mathbf{H}_{l-1} \cdot \mathbf{W}_l + \mathbf{b}_l). \quad (2)$$

Several non-linear activation functions have been proposed. Here we choose the Rectified Linear Unit (ReLU) [29] in all C layers and FC layers for its high capability and efficiency:

$$Actv(x) = \max(0, x). \quad (3)$$

The last classification layer is usually a softmax layer with the amount of neurons equaling the number of classes to be classified. We use a logistic regression layer with one neuron to do binary classification, which is similar to a FC layer except that the sigmoid activation function should be used:

$$Actv(x) = \frac{1}{1+e^{-x}} \quad (4)$$

The activation value represents the probability of the input belonging to the positive class.

The weights $\{\mathbf{W}_1,...,\mathbf{W}_L\}$ and the biases $\{\mathbf{b}_1,...,\mathbf{b}_L\}$ compose the model parameters, which are iteratively and jointly optimized through maximization of the classification accuracy over the training set.

2.2. Saliency detection using Mr-CNN

Inspired by [30-32], we develop a CNN architecture with multiple resolutions (or scales) to simultaneously learn early features, bottom-up saliency, top-down factors and their integration from image data for predicting eye fixations. Specially, we consider three properly designed resolutions. For the input layer, we extract image regions of fixed size centered on the same locations from images with different scales to form multiresolution inputs. We first rescale the input image to three scales by simply warping it directly and ignoring its original size and aspect ratio. Then

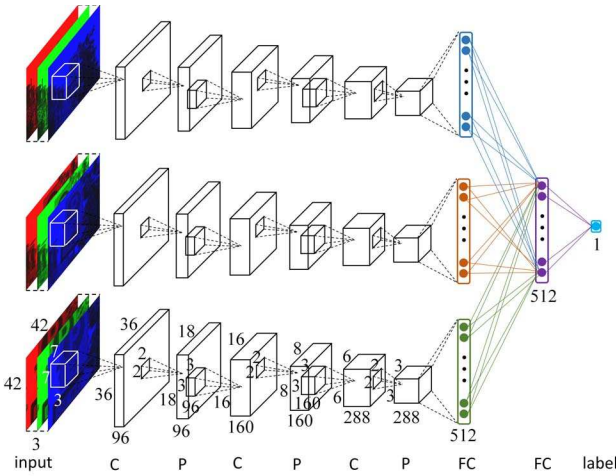

Figure 2: Network architecture of our Mr-CNN. Convolutional layer, max-pooling layer and fully connected layer are denoted as C, P and FC respectively. The sizes of the input image, feature maps, FC layers, convolution kernels and pooling windows are marked in the last stream of the Mr-CNN, which shares the same network architecture and the same parameters in C layers over 3 streams.

we extract image regions of same size at the same center location from the three rescaled image duplicates mentioned above. Thus the three image regions constitute the multiresolution architecture containing information flows with small-to-large contexts and coarse-to-fine granularities. In this paper, the three scales used to rescale the input image are empirically chosen as 150×150, 250×250, and 400×400, respectively. The size of image regions is set to 42 experimentally (see Figure 1 and Figure 2).

As for the network architecture, as shown in Figure 2, our Mr-CNN starts from three streams in lower layers. Each stream is composed of three C layers, three P layers, and a FC layer. Subsequently the three streams are fused using another FC layer, which is followed by one logistic regression layer at the end to perform classification. The three streams are separated shoulder to shoulder before the second FC layer and then are combined into one layer for jointly inferring the bottom-up saliency among the multi-resolution inputs. Here we share the parameters in each C layer across three streams to learn scale-invariant features. We use 96 filters with size 7×7 in the first C layer, 160 and 288 filters with size 3×3 respectively in the second and the third C layer. We set the convolution stride to 1 and perform valid convolution operations, disregarding the map borders. We also choose to use 2×2 pooling windows in all P layers, 512 neurons in all FC layers and 1 neuron in the output layer, resulting in the whole network of size I[42×42×3(×3)]-C[36×36×96(×3)]-P[18×18×96(×3)]-C[16×16×160(×3)]-P[8×8×160(×3)]-C[6×6×288(×3)]-P[3×3×288(×3)]-FC[512(×3)]-FC[512]-O[1] (see Figure 2), where we write the size and the attribute of each layer in and out of brackets respectively. The denotation (×3) means the layer has three duplicates in three streams. The input and the output layers are abbreviated as I and O respectively.

In the training stage, we randomly sample fixation and non-fixation locations based on the saliency values in the ground truth density maps which are generated by applying Gaussian blur on the raw eye fixation point maps. Then, we extract image regions centered at the sampled fixation or non-fixation locations as the inputs of our Mr-CNN, together with their corresponding binary classification labels. Here we consider fixation and non-fixation image regions as positive set and negative set respectively. Afterwards, we train the Mr-CNN using back propagation algorithm [33] and gradient descent algorithm based on the minimization of the cross entropy between the predicted labels and the ground truth labels in the last layer.

When testing, to reduce computation cost, we sample 2500 locations for each testing image as center locations to extract image regions, which is implemented by evenly sampling 50 locations along each side of the testing image. Then the activation value of the last layer in the Mr-CNN is obtained as the saliency value of each location to form the down-sampled saliency map. Ultimately, the obtained down-sampled saliency map is rescaled to the original size of the testing image to achieve the final saliency map.

3. Experiments

In this section, we report experimental results to evaluate the proposed approach in eye fixation prediction. We first introduce the eye fixation benchmark datasets and the evaluation metrics used in this paper, followed by the implementation details of our model. Then the results of our approach and comparisons with 11 state-of-the-art saliency models over four datasets are presented. Finally, the hierarchical features learned by the proposed Mr-CNN are visualized and some fatal parameters are analyzed.

3.1. Datasets

We conducted evaluation on four widely used eye fixation datasets with different characteristics. The first dataset, MIT [2], contains 1003 images collected from Flicker and LabelMe datasets, with resolution ranging from 405×1024 to 1024×1024 pixels. It is the largest eye fixation dataset and consists of 779 landscape, 228 portrait and several synthetic images free-viewed by 15 human subjects. The second dataset, Toronto [11], contains 120 color images of indoor and outdoor scenes with a fixed resolution of 511×681 pixels. These images are free-viewed by 20 human subjects. The third dataset, Cerf dataset [24], is made up of 181 images with resolution of 1024×768 pixels. The contents of interest in this dataset are usually faces and some other small objects like cell phones, toys, etc. Each image in this dataset is viewed by 7 subjects.

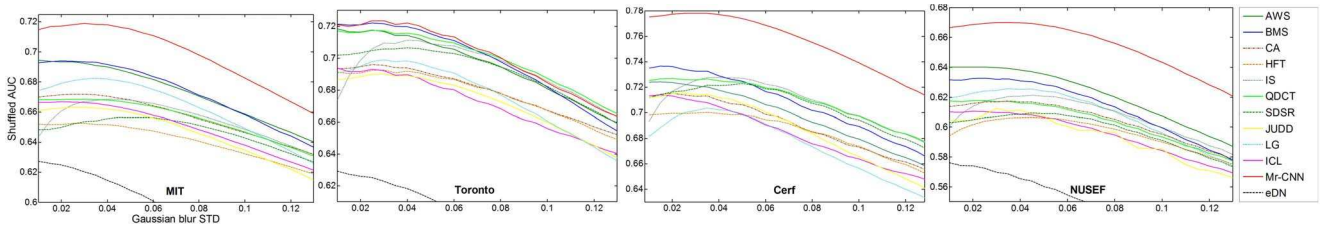

Figure 3: Qualitative model comparisons. Fixation prediction accuracy of our Mr-CNN model compared with 11 state-of-the-art models over 4 benchmark datasets. The result of eDN over the Cerf dataset is not shown. X-axis indicates the Gaussian blur STD σ (in image width) by which saliency maps are smoothed and Y-axis indicates the average shuffled-AUC score on one dataset.

Dataset	AWS	BMS	CA	eDN	HFT	ICL	IS	JUDD	LG	QDCT	SDSR	Mr-CNN
MIT	**.6945**	.6939	.6718	.6273	.6526	.6667	.6686	.6631	.6823	.6686	.6588	**.7190**
Opt. σ	.010	.020	.025	.010	.025	.020	.040	.025	.035	.025	.045	.030
Toronto	.7184	**.7221**	.6959	.6292	.6926	.6939	.7115	.6901	.6990	.7174	.7065	**.7236**
Opt. σ	.010	.025	.025	.010	.030	.010	.040	.030	.030	.025	.040	.030
Cerf	.7241	**.7367**	.7152	-	.7001	.7137	.7276	.7154	.7035	.7267	.7227	**.7781**
Opt. σ	.010	.010	.025	-	.035	.010	.035	.025	.035	.020	.050	.030
NUSEF	**.6403**	.6328	.6174	.5761	.6065	.6105	.6213	.6124	.6256	.6176	.6093	**.6702**
Opt. σ	.020	.025	.035	.010	.045	.020	.045	.030	.035	.025	.045	.035
Average	.6943	**.6964**	.6751	.6109	.6630	.6712	.6823	.6703	.6776	.6826	.6743	**.7227**

Table 1: Maximum performance of models shown in Figure 3. Optimal scores of each model over different datasets and the corresponding Gaussian blur STD are reported. The highest scores over the compared 11 models on each dataset are shown in bold face font, the highest ones over all models are both in bold face font and underlined. The result of eDN model over the Cerf dataset is not available. The average score of eDN is calculated over three datasets.

The last dataset, NUSEF [21], is a newly proposed dataset with 758 semantically-rich images containing affective contents such as expressive faces, interesting objects, and actions. On average, each image in this database is viewed by 25 subjects. In our experiments, we use 431 images in this dataset due to the copyright issue.

3.2. Evaluation metrics

One of the most widely used metrics to evaluate saliency models is the Area Under the ROC Curve (AUC) [11]. For an image, human eye fixation points are considered as positive set and non-fixation points are regarded as negative set. Then, the computed saliency map is binarily classified into salient region and non-salient region by a threshold. By varying the thresholds, ROC curve is achieved by plotting true positive rate vs. false positive rate, with its underneath area calculated as AUC score. However, AUC can be greatly influenced by center-bias [34] and border cut [35]. Consequently, it would generate a large value for a central Gaussian blob, leading to unfair evaluation. To cope with these issues, shuffled AUC is introduced by [34, 35]. Contrary to AUC, shuffled AUC adopts all fixation points (except for the positive set) over all images from the same dataset as the negative set. Using shuffled AUC, the score of a central Gaussian blob is 0.5 while the score of a perfect prediction is 1. Considering the sensitivity of the shuffled AUC score to different levels of blurring applied on saliency maps, we follow many recent works [4, 6] to smooth the saliency maps using small Gaussian filters with various standard deviation (STD) σ. Then we show the curve of average shuffled AUC scores over a datasets vs. various σ and report the best score under the optimal σ to evaluate a model.

3.3. Implementation details

Data processing. We did data augmentation by horizontally flipping each image to double image samples so as to enhance model generalization. During training, we sampled 10 fixation locations and 20 non-fixation locations per training image based on whether the corresponding saliency values in the eye fixation density maps are greater than 0.9 or smaller than 0.1. When testing, for an original testing image, we averaged its saliency map and the one of its horizontally flipped version as the final saliency map. When extracting image regions given the center locations, if the center pixel close to image borders, it will result in insufficient pixels to extract. In this situation, we copied image borders to form image regions with the same size. Before training, each dimension in the training image regions was mean-centered and normalized to unit variance over each training set, and the same normalization process was also used in the testing stage.

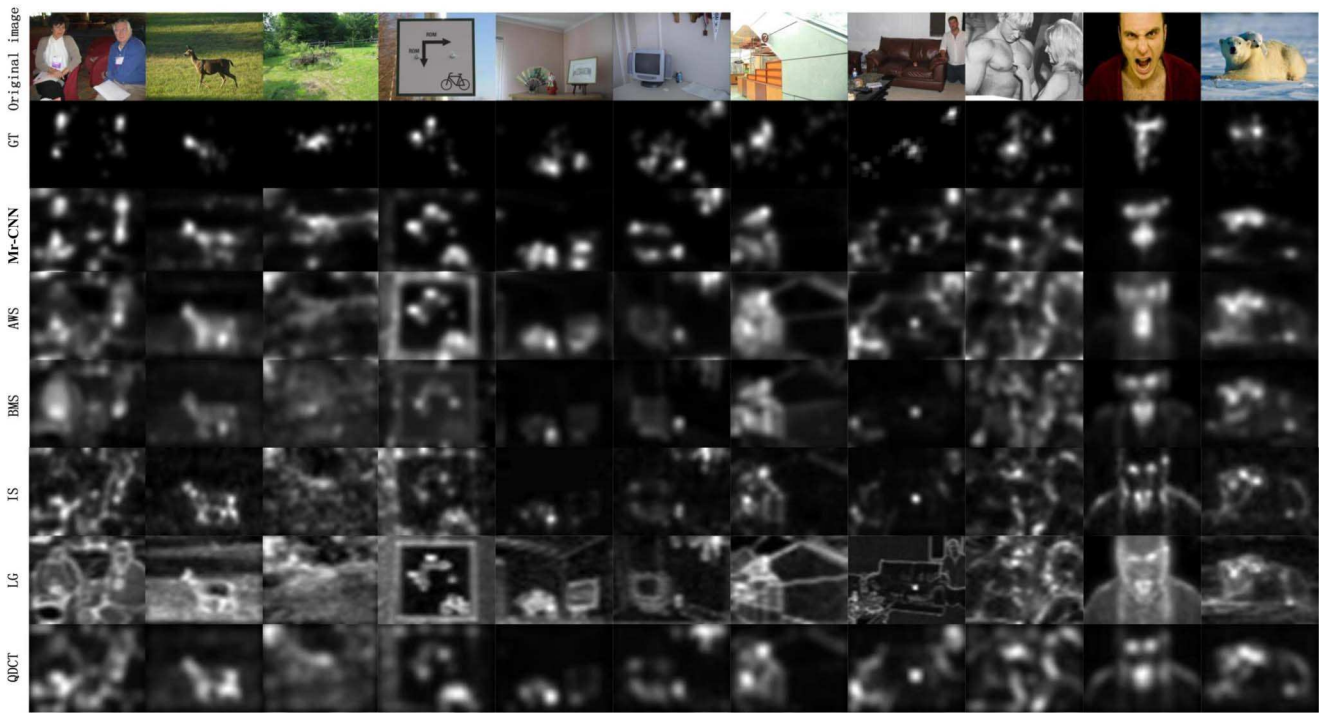

Figure 4: Visual comparisons of different models. We compare some saliency maps of our Mr-CNN model with other 6 models, i.e. AWS, BMS, IS, LG and QDCT, which perform best over 4 datasets based on the average shuffled AUC scores in Table 1. The first row shows the input images from the MIT (the first 4 columns), Toronto (the 5th to 7th columns), Cerf (the 8th column) and NUSEF (the last 3 columns) datasets. The second row shows the corresponding ground truth fixation density maps (GT) which are generated by applying Gaussian blur on the raw eye fixation point maps.

CNN parameters and settings. We trained and tested our model over each dataset using 10-fold cross-validation. Specifically, we averagely and randomly divided the dataset into 10 partitions. 9 partitions were used for training and the remaining 1 partition was used for testing. This was repeated such that each partition in the dataset is used once as the testing data. During the iterative process in training the Mr-CNN, we set the training step to 5,000 where one mini-batch was trained per step. Meanwhile we used 1/9 of the training set as the validation set to avoid overfitting. In details, we evaluated the performance of the Mr-CNN every 200 training steps, and selected the best trained network with the minimal cross entropy over the validation set. We set the size of mini-batch to 256 and 128 respectively for MIT dataset and other 3 datasets during training, with respect to the different image amount of these datasets. Besides, we used weight decay of 0.0002 and momentum linearly increased from 0.9 to 0.99 along with the increasing training step in all networks. To alleviate overfitting, dropout [36] was used with the corruption probability of 0.5 in the third C layer and the subsequent two FC layers for all networks. We also used a weight constraint [36] of 0.1 to the convolutional kernels of the first C layer so that once the ℓ_2-norm of a kernel is larger than the constraint, it could be renormalized by division. This also may relieve overfitting.

Transfer learning. Given that the MIT dataset contains the largest amount of images among the four datasets and consists of various salient contents, including both bottom-up and top-down ones, we utilized models trained on it to transfer domain knowledge to other three datasets to overcome the problem of lacking training images. We first trained the networks over MIT dataset with the learning rate initially set to 0.002 and subsequently decay along with the increasing training step. Then we simply adopted one of the networks trained in the 10-fold cross-validation process as the pre-trained network for the other three datasets, instead of training a new model using all MIT images. On other three datasets, the networks are fine-tuned given a relatively small learning rate initially set to 0.0001 and a smaller one fixed to 0.000001 respectively for the last 4 layers and the first 2 C layers, considering the low-level features can generalize well over natural scene images.

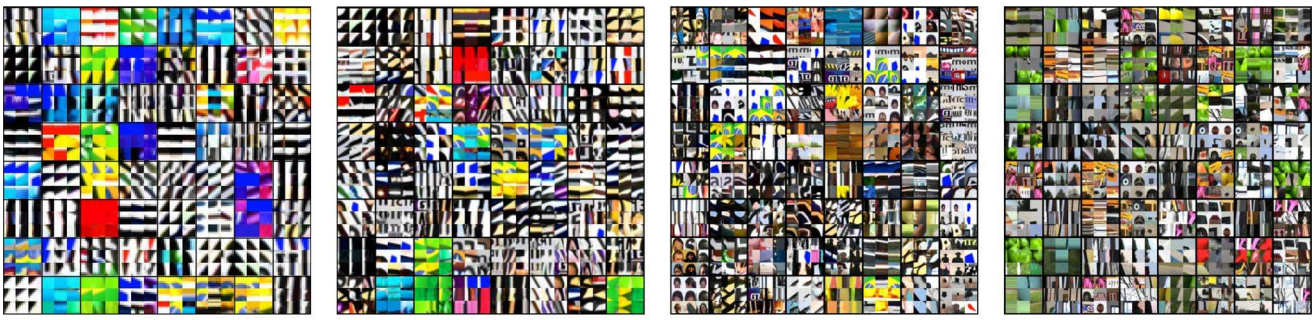

Figure 5: Feature visualization on MIT and Toronto datasets. Best viewed in digital version.

Platform and routine. The proposed model was implemented using Matlab, python and CUDA, run on a workstation with 2 2.8GHz 6-core CPUs, 32GB memory, 64-bit Windows sever 2008 OS, additionally with a GTX Titan black GPU for acceleration. The CNN routine we used is based on the *deepnet*[1] library. The average time taken to test an image is 14s.

3.4. Results

To demonstrate the effectiveness of the proposed Mr-CNN model in predicting eye fixations, we evaluated it by comparison to 11 state-of-the-art models, including AWS [5], BMS [6], CA [7], eDN [27], HFT [37], ICL [12], IS [16], JUDD [2], LG [4], QDCT [17], and SDSR [38] . These methods selected for comparison have been proposed in recent years and their codes or calculated saliency maps are publicly available[2]. We first evaluated the shuffled AUC scores over our model and other 11 models for quantitative comparison. The saliency maps were smoothed by Gaussian kernels with various blur STD σ first, then average shuffled-AUC scores of each model on different datasets over varying σ were presented in Figure 3. Optimal scores of each model over different datasets and the corresponding Gaussian blur STDs were reported in Table 1.

As shown in Fig. 3 and Table 1, the proposed Mr-CNN model achieves the best performance on all four benchmark datasets. Especially, it is significantly better than other 11 methods on the MIT, Cerf, and NUSEF datasets. On the Toronto dataset, our model is slightly better than other models. We presume that this is because the Toronto dataset contains relatively less images, which usually hurts the performance of deep learning models. From our comparison results, AWS and BMS ranked the second echelon. We also notice that in [26], authors adopted AUC as the metric and eDN method shows the best performance. However, its performance is not good using the metric of shuffled-AUC scores. It is well recognized that shuffled-AUC is a better metric to fairly compare different saliency models.

We also give the qualitative comparison of our model with other 6 best models in Figure 4. As we can see, our Mr-CNN model can detect not only bottom-up saliency patterns (e.g., Column 3, 5, 6, 7, 9), but also diverse top-down factors, such as faces (e.g., Col 1, 8, 9), text (e.g., Col 4, 5), animal heads (e.g., Col 2, 11), which are difficult for traditional methods.

3.5. Feature visualization

To further understand the learned Mr-CNN, we visualized the hierarchical features of the C layers learned on MIT dataset and the features of the third C layer learned on Toronto dataset. Considering the low-level features can generalize well over different natural images, we didn't visualize the features in lower layers learned over Toronto dataset. As it is difficult to visualize convolutional kernels in higher layers of CNNs, for each kernel we uniformly show 9 optimal stimuli which most strongly activate the corresponding neuron. We just show 64 kernels per layerfor space limitation, forming an 8×8 matrix (see Figure 5). As shown in Figure 5, our Mr-CNN mainly learns various edges and color blobs in layer 1, diverse corners and edge/color conjunctions in layer 2. The features learned in layer 3 are very informative. On MIT dataset, there contain many low-level patterns, for instance, complex corners ((Row 2, Col 7), and (Row 5, Col 2)), edge conjunctions ((Row 1, Col 3), (Row 6, Col 1), (Row 6, Col 2), (Row 6, Col 4) and so on), complex textures ((Row 4, Col 2) , (Row 5, Col 7), (Row 7, Col 6) and so on), and other contrast-like patterns ((Row 1, Col 7), (Row 3, Col 6), (Row 8, Col 8) and so on). These features are essentially related to bottom-up saliency. Meanwhile, we can see on MIT dataset, layer 3 also learns some high-level semantic concepts, for instance, eyes or eye-like patterns ((Row 4, Col 6)), faces ((Row 7, Col 1)), human heads ((Row 8, Col 1)), human body profiles or similar patterns ((Row 7, Col 8)), and text ((Row 2, Col 8), (Row 3, Col 8), (Row 4, Col

[1] https://github.com/nitishsrivastava/deepnet

[2] The authors of the eDN model only published their saliency maps on three datasets, i.e. MIT, Toronto, and NUSEF. Thus we didn't evaluate the eDN model on Cerf dataset.

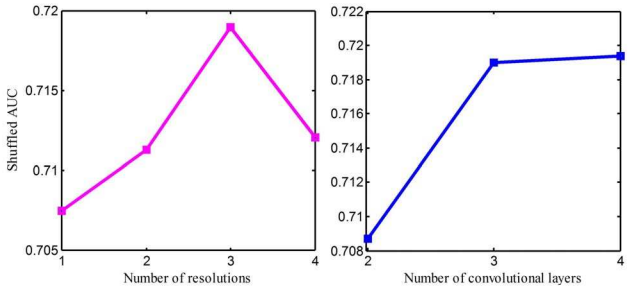

Figure 6: Network structure analysis on MIT dataset. (a): Effect of number of resolutions. We rescaled images to 250×250 when testing one resolution, 150×150 and 250×250 when testing 2 resolutions, 90×90, 150×150, 250×250 and 400×400 when testing 4 resolutions. (b): Effect of number of convolutional layers. We just abandoned the third C layer from our model when testing 2 C layers. When testing 4 C layers, we just got rid of the last pooling layer and added a C layer with 384 3×3 kernels above. The model with 3 resolutions and 3 C layers is the one we used in this paper.

8)). This indicates that our Mr-CNN can learn both bottom-up saliency cues and high-level top-down factors. On Toronto dataset, layer 3 mainly learns many low-level and mid-level patterns. It seems it fails to learn much semantic concepts as this dataset mainly consists of diverse plain objects and lacks obvious semantic contents.

3.6. Network structure analysis

Here we analyze how the network structure influences the model performance. We mainly tested two fatal factors, namely, the number of resolutions we used and the number of convolutional layers, on MIT dataset. As shown in Figure 6(a), when we increase the number of resolutions, the model performance goes up first, then reaches the peak when three resolutions are used as in our model, subsequently drops down. As for the effect of different numbers of C layers, considering it's too naive to just use one C layer in a deep convolutional network, we just additionally test our model with 2 and 4 C layers. As shown in Figure 6(b), increasing the number of C layers from 2 to 3 boosts the model performance apparently, then the model performance nearly saturates. Although using 4 C layers can still enhance our model performance a little, it also increases much more training and testing time. Thus we adopted three C layers regarding the tradeoff between model capability and computation cost.

4. Conclusions and future works

In this work, we have proposed a novel convolutional neural network based eye fixation prediction model. Our model has achieved the best performance with significant improvement to 11 state-of-the-art saliency models on four publically available benchmark datasets. The superior performance of our method indicates that the human visual system is more likely to process low-level contrast and high-level semantics jointly rather than separately. The learned hierarchical features were visualized to show that our Mr-CNN learns both low-level saliency cues and high-level factors. The above results demonstrated that the proposed model can obtain promising performance by simultaneously learn early features, bottom-up saliency, top-down factors, and their integration directly from image data. More importantly, the proposed model architecture can also help to improve our understanding of the internal mechanism of fixation selection in the human visual system.

In the future, we will further extend the proposed work in two aspects. First, we can explore the effect of each saliency related feature uncovered in the visualization experiment section, this may offer novel insights for the understanding of human vision system. The second aspect is to extend our model to predict eye fixations while viewing video sequences.

Acknowledgements: This work was partially supported by the National Science Foundation of China under Grant 61473231.

References

[1] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254-1259, 1998.

[2] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV,* 2009.

[3] A. Borji. Boosting bottom-up and top-down visual features for saliency estimation. In *CVPR,* 2012.

[4] A. Borji, and L. Itti. Exploiting local and global patch rarities for saliency detection. In *CVPR,* 2012.

[5] A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil. Saliency from hierarchical adaptation through decorrelation and variance normalization. *Image Vision Comput.*, 30(1): 51-64, 2012.

[6] J. Zhang, and S. Sclaroff. Saliency detection: A boolean map approach. In *ICCV,* 2013.

[7] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10):1915-1926, 2012.

[8] F. Liu, and M. Gleicher. Video retargeting: automating pan and scan. In *ACM Multimedia,* 2006.

[9] J. Sun, and H. Ling. Scale and object aware image retargeting for thumbnail browsing. In *ICCV,* 2011.

[10] J. Han, K. Li, L. Shao, X. Hu, S. He, L. Guo, J. Han, and T. Liu. Video abstraction based on fMRI-driven visual attention model. *Information Sciences*, 2014.

[11] N. Bruce, and J. Tsotsos. Saliency based on information maximization. In *NIPS,* 2005.

[12] X. Hou, and L. Zhang. Dynamic visual attention: Searching for coding length increments. In *NIPS,* 2009.

[13] B. Han, H. Zhu, and Y. Ding. Bottom-up saliency based on weighted sparse coding residual. In *ACM Multimedia,* 2011.

[14] X. Hou, and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR,* 2007.

[15] C. Guo, and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Trans. Image Process.,* 19(1):185-198, 2010.

[16] X. Hou, J. Harel, and C. Koch. Image signature: Highlighting sparse salient regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(1):194-201, 2012.

[17] B. Schauerte, and R. Stiefelhagen. Quaternion-based spectral saliency detection for eye fixation prediction. In *ECCV*, 2012.

[18] Q. Zhao, and C. Koch. Learning a saliency map using fixated locations in natural scenes. *J. Vision,* 11(3):9, 2011.

[19] W. Einhäuser, M. Spain, and P. Perona. Objects predict fixations better than early saliency. *J. Vision,* 8(14):18, 2008.

[20] L. Elazary, and L. Itti. Interesting objects are visually salient. *J. Vision,* 8(3):3, 2008.

[21] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T.-S. Chua. An eye fixation database for saliency detection in images. In *ECCV*, 2010.

[22] R. Carmi, and L. Itti. Visual causes versus correlates of attentional selection in dynamic scenes. *Vision Research*, 46(26):4333-4345, 2006.

[23] B. W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *J. Vision*, 7(14):4, 2007.

[24] M. Cerf, J. Harel, W. Einhäuser, and C. Koch. Predicting human gaze using low-level saliency combined with face detection. In *NIPS*, 2008.

[25] C. Shen, M. Song, and Q. Zhao. Learning high-level concepts by training a deep network on eye fixations. In *NIPS Deep Learning and Unsup Feat Learn Workshop*, 2012.

[26] E. Vig, M. Dorr, and D. Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *CVPR*, 2014.

[27] Y. Lin, S. Kong, D. Wang, and Y. Zhuang. Saliency detection within a deep convolutional architecture. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278-2324, 1998.

[29] V. Nair, and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[31] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1915-1929, 2013.

[32] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. MIT Press, Cambridge, MA, USA, 1986.

[34] B. W. Tatler, R. J. Baddeley, and I. D. Gilchrist. Visual correlates of fixation selection: effects of scale and time. *Vision Research*, 45(5):643-659, 2005.

[35] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. SUN: A Bayesian framework for saliency using natural statistics. *J. Vision*, 8(7):32, 2008.

[36] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv*:1207.0580, 2012.

[37] J. Li, M. D. Levine, X. An, X. Xu, and H. He. Visual saliency based on scale-space analysis in the frequency domain. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(4):996-1010, 2013.

[38] H. J. Seo, and P. Milanfar. Static and space-time visual saliency detection by self-resemblance. *J. Vision*, 9(12):15, 2009.

[39] J. Han, S. He, X. Qian, D. Wang, L. Guo, and T. Liu. An object-oriented visual saliency detection framework based on sparse coding representations. *IEEE Trans. Circuits Syst. Video Technol.,* 23(12):2009 -2021, 2013.

[40] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu. Background prior based salient object detection via deep reconstruction residual. *IEEE Trans. Circuits Syst. Video Technol.,* 2014. DOI: 10.1109/TCSVT.2014.2381471.