Efficient Visual Attention with Deep Learning

Anonymous CVPR submission

Paper ID ****

Abstract

Visual saliency is an important component of attention. It helps animals survive and can also be used by computer vision applications to filter out irrelevant information from high volumes of data. In this work, we present a new fully convolutional neural network architecture designed for detecting visual saliency. We also propose methods of data pre-processing that is specifically beneficial for the task of visual saliency detection. Experiments carried out with the MIT300 benchmark presented state-of-the-art performance and a parameter reduction of 3/4 compared to similar models.

1. Introduction

One of the most challenging unsolved problems in Artificial Intelligence is vision. However, it is fundamental for the conception of systems that interact in the real physical world. Such systems would be useful for applications in areas like domestic services, industry, and agriculture, with great potential for the benefit of society.

Vision is remarkable data and computationally intensive. In humans, approximately half of the brain is involved in vision-related tasks [4]. Even our minds cannot handle all the sheer amount of sensorial information received every second. In order to deal with this amount of data, humans have attentional systems, a fundamental mechanism that, among other functions, filters out irrelevant information – either visual or from other senses– and helps us focusing our cognitive processes on what is important at a given moment. These facts are a strong evidence that in order to help to solve vision problems, attention should be applied.

Visual attention can be defined as the delimitation of a certain spatial the region on an image for further cognitive processing [12]. The phenomenon emerges from two fundamentally different processes: the *top-down* mechanism that implements our longer-term cognitive strategies by biasing attention according to one's interests (e.g. find a red apple in a tree because of hunger, which will make red be more recognizable on the scene), and the *bottom-up* mechanism [2], a process generated through external stimuli that captures one's attention from its conspicuousness level. In this work, we focus on the latter, also named visual saliency.

(a) (b)

Figure 1. Example of visual saliency. b) is the saliency map where brighter pixels (warmer colors) represent regions more salient to humans on the original image a).

Visually salient regions on images are usually represented by *saliency maps* (Figure 1). In these maps, images are generated such that areas with high-valued pixels express high saliency on the original image, whereas regions with low-valued pixels represent low saliency. Datasets with such maps are obtained by collecting eye-fixation data from humans while observing the scenes.

1.1. Related work

Early computational models of visual saliency were generally built based on filtering of images for extraction of a pre-selected set of features considered important for *bottom-up* attention. *Vocus* [5] is a computational model that extracts features are shown to be naturally salient to humans such as color/luminance contrast and orientation from different scales of the image.

A rapid change of paradigm occurred around 2015 when *Deep Learning* techniques showed to be very effective in the generation of saliency maps. *Salicon* [6] demonstrated that the use of convolutional neural networks with weights initialized from image classification networks, e.g. *VGG-16* [10] could considerably increase the similarity of computed maps to those generated from humans. *Salicon* uses different scales of the image as input to capture relevant information in the context of saliency, using the same network weights for each dimension. *ML-Net* [3] uses the output of different layers of *VGG-16*, combining them in many dimensions and various levels of abstraction. *DeepFix* [8] extends a pre-trained model with inception [11] layers – which use information from different scales of the image – and a component for center bias, a phenomenon that arises from our tendency to take pictures with relevant objects at the center of the image. *Salnet* [9] explores two models: a shallow convolutional network followed by a fully-connected layer and a fully convolutional neural network with first layers' weights initialized from *VGG-16*.

Visual saliency detection models are usually evaluated and ranked on *MIT saliency benchmark* [1], which uses a variety of metrics to express how close generated saliency maps are to those created from human data.

However, current state of the art models are in general quite expensive computationally, partly because most of them are based on big pre-trained networks. The convolutional layers of *VGG-16* are composed of around 14.7 million parameters. While pre-trained weights from classification tasks showed to be effective for saliency prediction, it is reasonable to question whether creating a proper network from scratch could yield a smaller amount of parameters that are more efficient for the sole task of saliency prediction. Also, there are some data processing methods from previous work on psychology-based models that were not used in current models but are considered worthwhile to explore, such as using global information from the scene and a colorspace more closely related to human vision.

This work aims at building a visual saliency model that is a) effective, yielding results similar to another state of the art models, and b) relatively simple and computationally efficient. It is important that both criteria are matched in order to extend the model in the future for video and real-time computer vision applications such as navigating robots.

2. Proposed model

Figure 2 shows the overall architecture of the fully convolutional neural network proposed in this work. It extracts features from increasingly smaller dimensions of the input image. The network is composed of four main blocks:

1. The first level extracts low-level features from the input image, of dimensions $W \times H \times 3$ (width, height, depth), using a single layer with 48 convolution filters with ReLu activation followed by max-pooling that reduces the image by a factor of two. It was found that further decreasing the number of filters in this layer considerably hurts performance, which makes sense because it is important to capture high spatial frequency and high contrast information in the context of visual saliency.
2. The second level extracts low-medium level features from the input of dimensions $W/2 \times H/2 \times 48$ using two layers with 74 and 96 convolution filters, respectively, followed by ReLU activation and max-pooling.
3. The third level extracts medium-high level features from input with dimensions $W/4 \times H/4 \times 96$ using four convolution layers in sequence with 128, 144, 160 and 172 filters. Every convolution layer is followed by ReLu. Max-pooling is carried out at the end. A considerable depth in this level was found to be important for the network's performance.
4. The fourth and last level is composed of eight inception blocks that extract high level features from the input with dimensions $W/8 \times H/8 \times 172$. Great level of depth and Inception blocks were found to be very important at this level. A 1×1 convolution makes a linear combination of the output maps at the end of the 8 inception blocks, followed by ReLu, producing the final saliency map of dimensions $W/8 \times H/8 \times 1$. The saliency map is then resized to the original dimensions using bicubic interpolation.

Figure 3 illustrates the inception architecture used. Each inception layer applies 3×3 convolution, a sequence of two 3×3 convolutions (these two types are preceded by 1×1 convolutions in order to reduce the number of input filters), 1×1 convolution, and a max-pooling of size 3×3. Each of those operations is executed in parallel from the same input and the outputs are concatenated at the output. Inception allows the network to use information from different spatial dimensions as well as previous layers (lower level saliency information) in the final map computation, which is considered to be important for visual saliency. The network has a total of 3593842 parameters, a very low number compared to other models. Table 2 details the filter configuration for the inception layers.

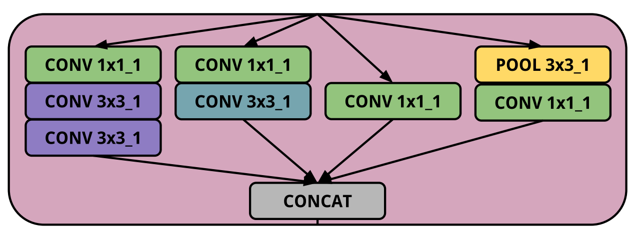


Figure 3. Inception block layout.

2.1. Data pre-processing

Input images were resized to dimensions $320 \times 240 \times 3$. Each image is normalized channel-wise by the subtraction of the channel mean and division by the standard deviation:

$$C = \frac{C - \mu_C}{\sigma_C}$$

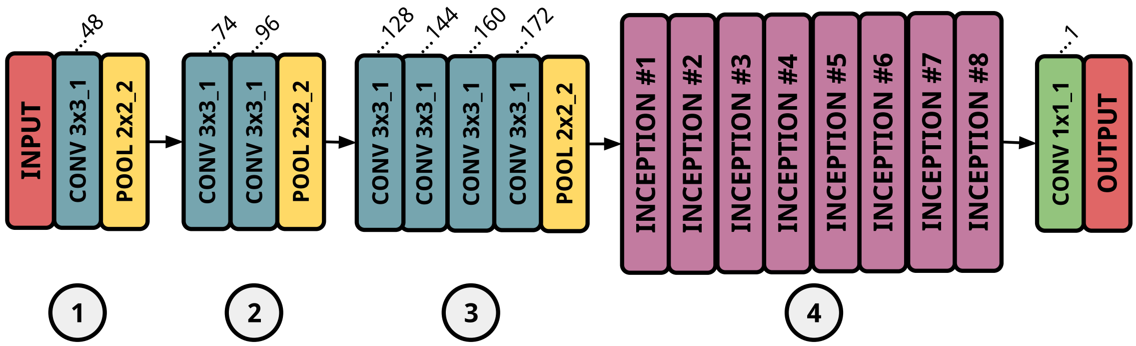


Figure 2. Overview of the network. Filters sizes are in format width×height_stride.

Table 1. Number of filters used in each inception block.

Block	pool	1×1	3×3 reduce	3×3	double 3×3 reduce	double 3×3
1	64	128	80	160	24	48
2	64	128	80	160	24	48
3	64	128	80	160	24	48
4	64	128	96	192	28	56
5	64	128	96	192	28	56
6	64	128	112	224	32	64
7	64	128	112	224	32	64
8	112	160	128	256	40	80

Most models used for comparison and cited in this work use RGB images normalized channel-wise using statistics computed from the dataset. However, visual saliency is highly connected to the context of the image, hence saliency depends on the local context.

Images are converted from RGB color space to the LAB colorspace. *Vocus* [5] cites that the LAB color space is more closely related to human vision once it encompasses red-green, yellow-blue and luminance maps.

Section 3.2 describes experiments carried out to investigate how different data pre-processing techniques affect the performance of the model.

2.2. Implementation

The network was implemented using *Tensorflow* `1.3.0` on a machine with *Ubuntu 16.04 LTS* and kernel *Linux* `4.8.0-54-generic`. Training was conducted on a GPU *NVIDIA GTX 1080* and the code is available at `https://goo.gl/WzpyYJ`.

2.3. Training

Two datasets were used: *SALICON* [6], with 15000 images, and *Judd* [7], with 1003 images. Data augmentation was applied by flipping images horizontally and vertically and applying random disturbances to the image: blur, adding noise, shifting and shearing. The cost function to be minimized was the *Mean Squared Error* between the ground-truth saliency map G and the predicted map P, using the *Adam Optimizer*.

Training iterated calculating the mean loss on the validation set after each epoch, and it stopped when the validation loss started to increase. *SALICON* dataset was first used. The number of images was 12000 in the training and 1500 in the validation set. Training iterated for 12 epochs with learning rate of 1×10^{-4} and batch size of 24 Finally, *Judd* was used, with 800 images in the training set and 203 images in the validation set. Training iterated for 3 epochs with learning rate of 1×10^{-5} and batch size of 10. The whole training process took around 4 hours.

3. Experimental Results

3.1. Results on MIT300 benchmark

Prediction took an average time of 10 milliseconds. Figure 4 shows some maps generated by the proposed model. They are generally quite similar to the ground truth. For evaluation, we submitted the model to the *MIT300 saliency benchmark* that has around 300 images and is commonly used to rank such models. Table 3 shows the resulting values for the most common metrics. The proposed model achieved results comparable to those of the state of the art while having, at least, one-fourth of the number of parameters.

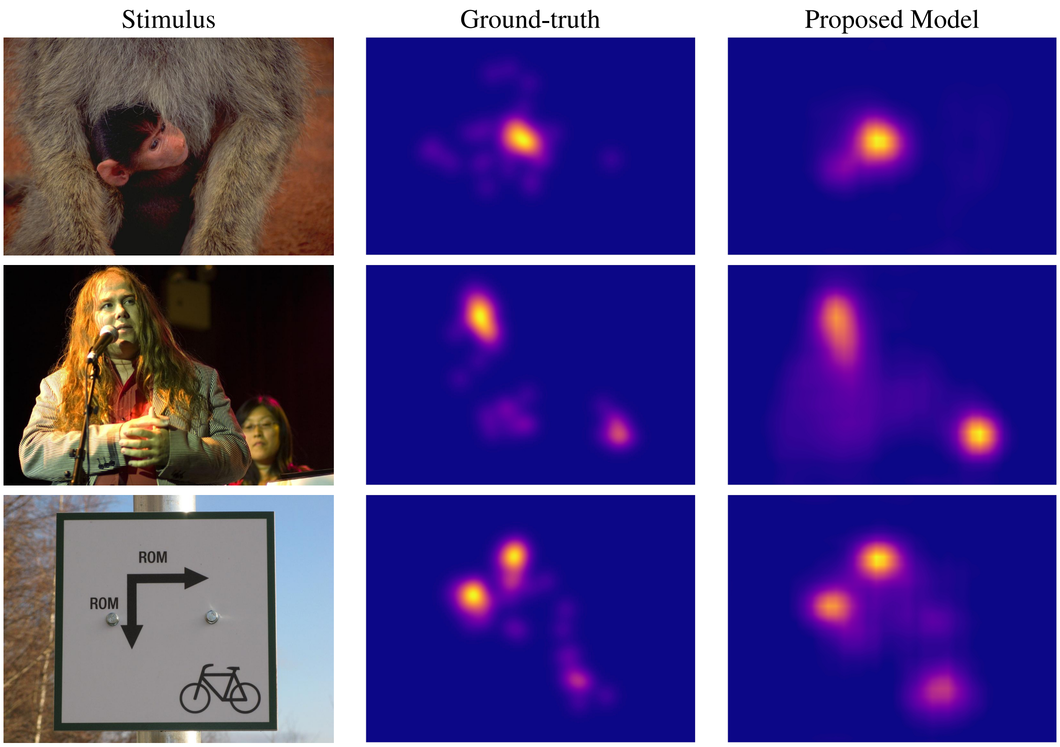


Figure 4. Examples of predictions made by the proposed model.

Table 2. State of the art models and metric scores on *MIT300 benchmark*.

Model	Num. parameters	AUC-Judd ↑	CC ↑	NSS ↑	Sim ↑	EMD ↓
Baseline: Infinite humans	-	0.92	1.0	3.29	1.0	0
DeepFix	≈16.7 million	0.87	0.78	2.26	0.67	2.04
Salicon	≈14.7 million	0.87	0.74	2.12	0.60	2.62
Proposed Model	**≈3.6 million**	**0.85**	**0.71**	**1.98**	**0.62**	**2.37**
ML-Net	≈15.4 million	0.85	0.69	2.07	0.60	2.53
SalNet	≈25.8 million	0.83	0.57	1.51	0.52	3.31

3.2. Comparison of data processing methods

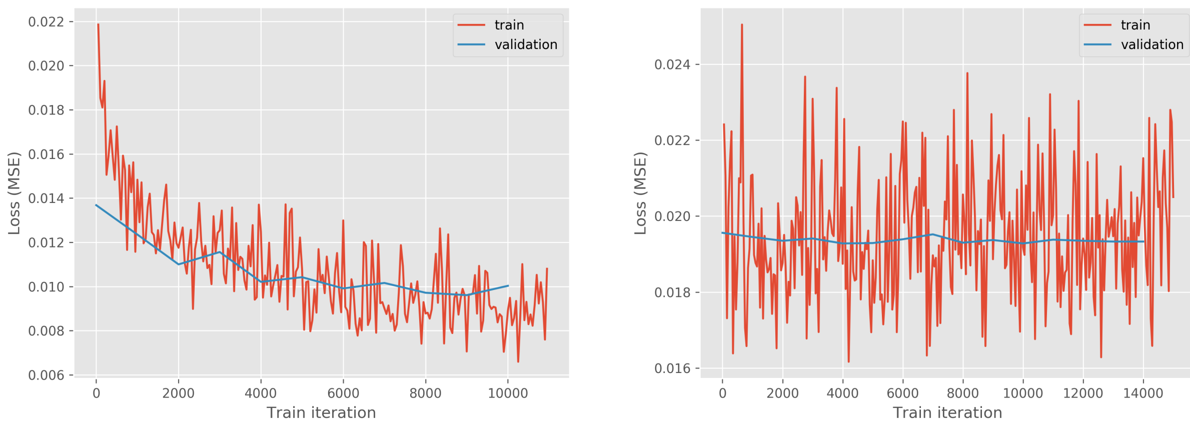
In order to investigate how the different data preprocessing methods affected model's performance, a comparison was made using two configurations: 1) Using LAB color space and normalizing channel-wise with mean and standard deviation per image; and 2) using RGB color space, normalizing channel-wise with a mean and standard deviation of the training dataset. The latter is commonly used in other models cited in this work. For each configuration, the model was trained on the *SALICON* dataset, which as split into 12000, 1500 and 1500 samples for training, validation, and test, respectively.

Figure 5 shows that configuration 1) converged faster and to a better optimum value. Figure 6 shows metrics computations on the test set. Overall, configuration 1) was better. We conjecture that the LAB color space in combination with normalization per image facilitates extraction of important features for the context of saliency: luminance and color contrasts and information on the global level of the image.

4. Conclusion

In this paper, we proposed a novel fully convolutional neural network for the prediction of visual saliency on images. The proposed model architecture was designed specifically for the task of saliency prediction. Normalizing images channel-wise per image – rather than per dataset – showed to be important for a good performance of the model. Our methods showed to be effective, yielding to a network with performance on *MIT300 benchmark* consistently among the ten best results on various metrics while having around $3/4$ fewer parameters than another state of the art models.

CVPR
#****



(1) (2)

Figure 5. Progression of losses on test and validation sets on configuration 1) (left) and 2) (right) during training.

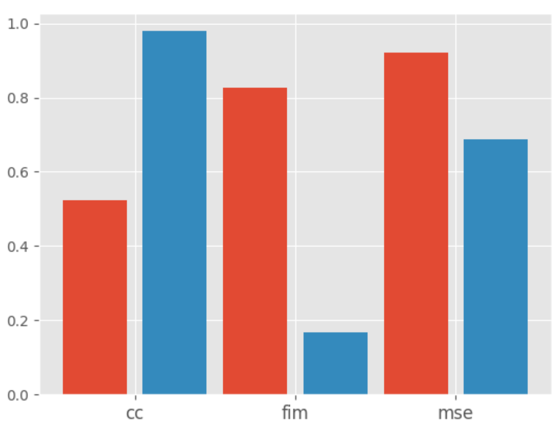


Figure 6. Metrics evaluation on the test set for both configurations.

References

[1] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. Mit saliency benchmark. http://saliency.mit.edu/index.html, 2016. 2

[2] E. L. Colombini. *An Attentional Model for Intelligent Robotics Agents*. PhD thesis, 2014. 1

[3] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. A deep multi-level network for saliency prediction. *arXiv preprint arXiv:1609.01064*, 2016. 2

[4] R. S. Fixott. Evaluation of research on effects of visual training on visual functions. *American Journal of Ophthalmology*, 44:230–236, 1957. 1

[5] S. Frintrop. Vocus: a visual attention system for object detection and goal-directed search. In *IN LECTURE NOTES IN ARTIFICIAL INTELLIGENCE (LNAI)*. Springer, 2005. 1, 3

[6] M. Jiang, S. Huang, J. Duan, and Q. Zhao. Salicon: saliency in context. *CVPR*, 2015. 1, 3

[7] T. e. a. Judd. Learning to predict where people look, 2016. 3

[8] S. S. S. Kruthiventi, K. Ayush, and R. V. Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *arXiv preprint arXiv:1510.02927*, 2015. 2

[9] J. Pan, E. Sayrol, X. G. i Nieto, K. McGuinness, and N. OConnor. Shallow and deep convolutional networks for saliency prediction. *arXiv preprint arXiv:1603.00845*, 2016. 2

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 2

[12] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognit Psychol*, 1980. 1