Determination of the eye-catching parts in graphical interfaces
in Proceedings of IIT.SRC 2017

Patrik BEKA*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
patrik.beka@gmail.com

Abstract. Eye-catching and graphically attractive design is an extremely important part of every website, although it might not appear to you at first sight. Our main purpose is to develop a neural network, capable of learning from given data and eventually predicting eye-catching and important parts of the websites. We selected method based on the convolution neural network, which is able to make predictions from the given images of the web pages. Final prediction is shown in the form of the heatmap, which determines the most engaging parts of the given web pages.

1 Introduction

Design of the good graphical interface for web pages is not an easy thing to do at all. Nowadays, when people are looking for informations almost only via internet, the good design has even bigger importance. Whether the visitor of page is interested in the content or not, is not as important as his attraction to the design of web page and the fundamental availability of informations. Therefore, design of the web page is certainly one of the most important preconditions for the future comeback.

Our task is to develop the system, that is capable of determining the eye-catching parts of web pages just from the web pages image (screenshot). In this thesis, we try to describe our method of determining important parts of the web pages using neural network, specifically convolution network, along with other similar solutions.

2 Related work

Convolution neural networks [7] [2] are widely used almost in every domain, where is the need for image recognition. Whether it is automatic face detection on Facebook, autonomous cars capable of self driving using autopilot by Tesla, Google, or software for sorting and classification of cucumbers in the Japanese farm[1]. Prototype of self driving car by Google called Dave-2 [1] has a model of convolution neural network, which processes frames from cameras placed on the car. These cameras are recording environment in 10 frames per second and after some modifications, these frames are passed to the neural network. This network consists of 9 layers, one normalization layer, 5 convolution and 4 fully-connected layers.

Related work about judging the importance of different parts in a web page can be classified into two main approaches.

First approach

First approach is using HTML code of the web page to divide the whole web page to the main parts with operation called segmentation. Segmentation uses either segmentation algorithms to divide page into the blocks by different features, or data structures to represent components and elements of HTML document. Most commonly used approaches to segmentation are:

- DOM-based segmentation:

  HTML document is represented as a DOM [4] (Document Object Model) tree. Tags are representing block of pages, for example P-paragraph, TABLE-table. Very accurate representation of

[*] Bachelor study programme in field: Informatics
Supervisor: Ing. Márius Šajgalík, Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

[1] https://cloud.google.com/blog/big-data/2016/08/how-a-japanese-cucumber-farmer-is-using- deep-learning-and-tensorflow

the structure of HTML document, but not accurate enough for dividing of visual different blocks.

- Location-based segmentation [10]:

  Page is divided to 5 parts: center, left, right, bottom, top. Problematic may be scrollable pages, when page has different layout of the objects after scrolling.

- VIPS (Vision-based Page Segmentation [3]) algorithm:

  Combination of the previous two examples of segmentation. Dividing page by color, size of blocks, etc. At first, the appropriate nodes, which imply the horizontal and vertical lines of web page, are found in the DOM tree. Based on that, semantic tree is created. It is a tree, where every segment is a separate node. Continuity of the segmented page is controlled by permitted degree of coherence (pDoC [9]), which ensures keeping content together, while semantically different blocks apart.

After segmentation, the importance of blocks is determining. It could be done using heuristics [8], human assessors (who manually label the blocks), etc.

As an example of working solution can be mentioned the work of Microsoft researchers [10]. They used VIPS algorithm for page segmentation and 5 human assessors to label each blocks of web pages. Blocks were numbered by importance from 1 to 4, from insignificant informations (such as adds) to the most important part of web page (news, products, etc.). Authors of this solution presume that people have consistent opinions about the importance of the same block in a page. After every block was labeled, the model of importance was created as a mapping function of every block and its importance:

$$\{block\ features\} \rightarrow \{block\ importance\} \quad (1)$$

For estimation of the block‘s importance was used neural network, where blocks are represented as tuple $\{x, y\}$. Number of block is x and its importance is y (real number). The network type was RBF (Radial Basis Function) and the neural network used standard gradient descent.

Second approach

Second approach to determining important parts of the web pages is based on the image of web page and a neural network that predicts the most engaging parts of web page in the form of heatmap, saliency map, etc.

Solution using this approach was created by Shen Chengyao and Zhao Qi [11]. They created a dataset divided to the categories by the content of web page (pictorial, text, mixed). Each category contains 50 images, which were shown to the 11 subjects. Their sequences of views were recorded using MATLAB with Psychtoolbox and Eyelink 1000. For views prediction was used MKL (multiple kernel learning), which was trained as a binary regression problem.

3 Method of predicting eye-catching parts of the websites

After closer research of the problem domain, we decided to use second approach of the determining important parts of the web pages. Our method is based on the neural network, particularly convolution neural network, which is able to predict the eye-catching parts just from the image of web page. After experimenting with various architectures of neural networks, we decided to use architecture shown in figure 1, because it has the best results. Shown architecture contains one convolution layer and one max pooling layer followed by normalization, fully-connected and output layer.

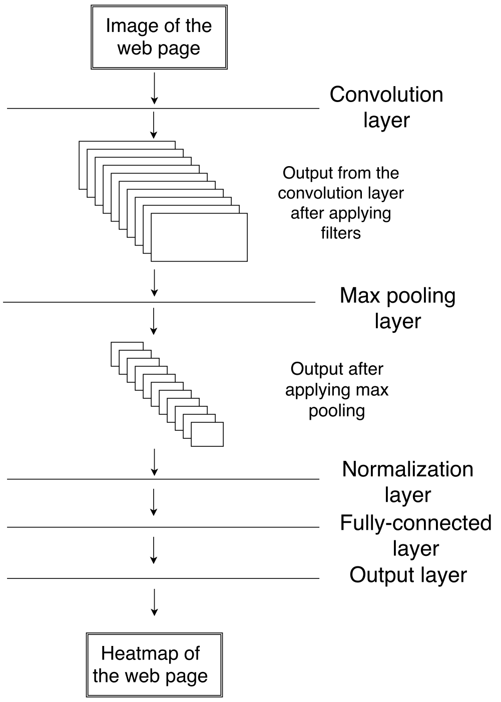


Figure 1. Diagram of the neural network‘s architecture

Convolution layer has convolution filter, whose size is 5x5. Activation function on this layer is standard ReLU:

$$f(x) = max(0, x) \quad (2)$$

After processing data, data is passed to the max pooling layer, where the output from previous layer is processed using window, whose size is 2x2, and MAX operation. Output of all filters is merged into one wide flat layer, normalization layer. This layer is followed by fully-connected and output layer, where the whole prediction is made. The output layer contains final predicted heatmap for image, whose size is *m* x *m*, where *m* is the size of both, image and heatmap.

4 Evaluation

Our dataset was collected as a part of experiments, that are described in bachelor thesis of Mária Dragúňová [5]. Now it contains 15 images of different web pages with 44 sequences of views (fixations) on these images in first 5 seconds.

Images size is *1920 x 1080*, what is too much. Therefore, the size is reduced to *256 x 256*, what is also better for neural network.

Fixations consist of *X* and *Y* coordinates for every view (range from 0 to 1) and duration of views. Number of fixations is variable for every image, the range is from 3 to 20. Heatmaps are calculated from these fixations using normal (Gaussian) distribution. First of all, every fixation is converted into the point on image of web page, as it is shown in equation 3. That ensures same size of heatmaps and images.

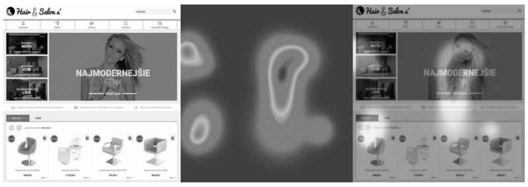
$$[x, y] = [fixation\_x * 256, fixation\_y * 256] \quad (3)$$

The probability density function of normal distribution is shown in equation 4, and it is calculated for every fixation from all other fixations.

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} * e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4)$$

x - current point for which the normal distribution is used
$(x-\mu)^2$ - distance between current point and other fixation
σ^2 - duration of fixation

For visualization of heatmaps on images, simple alpha blending is used, as you can see in figure 2.



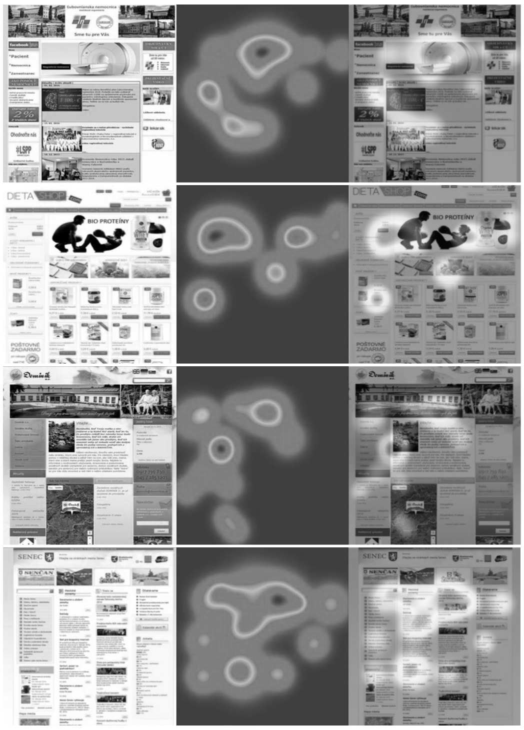


Figure 2. From left: original image, heatmap, heatmap on image

Before passing data to the convolution neural network, we divide the data (images, heatmaps) to 16 smaller parts (buckets), so every image and heatmap is basically *4 x 4* grid (shown in figure 3), consisting of squares of size *64 x 64*. This division of data is done mostly because of the need for a larger training dataset.

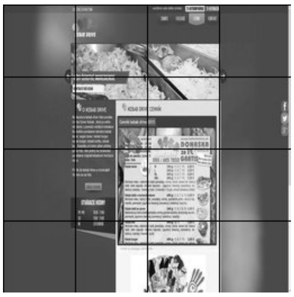


Figure 3. Visualization of the grid on image

The input data for neural network are parts of divided images, labels are equally divided heatmaps. Currently, accuracy of predictions is about 20%, what is not much.

5 Conclusion

This thesis describes a prototype of neural network, that is capable of predicting the eye-catching parts of the given web pages in form of the heatmap. It does not need any source code of the web sites, only screenshot is required. That may be considered as a solid advantage compared to some other solutions of similar problem.

Prototype is written in Python using Tensorflow[2] framework for neural networks. Results will be compared to the saliency model of Itti and Koch [6], but only after some modifications and more training. In future, we should consider optimization for network, maybe try to add more hidden layers. We may also perform another experiment with different web pages and let more people look at them, analyze it so we will have more data for training our convolution neural network and hopefully the dividing of data to 4x4 grid wouldn‘t be needed.

As a real-life application of our solution, I could imagine a web page, where people just upload screenshot of the web pages design and get a visualization of predicted engaging parts in form of the heatmap. That could provide solid informations for administrators of web sites, so they would be able to decide where to place important elements and where adds should be placed in order not to cause distraction.

Acknowledgement: This contribution was created with kind support of ČSOB Foundation and is partial result of the project University Science Park of STU Bratislava, ITMS 26240220084, co-funded by the ERDF.

References

[1] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to End Learning for Self-Driving Cars. *arXiv preprint arXiv:1604.07316*, 2016.

[2] Britz, D.: UNDERSTANDING CONVOLUTIONAL NEURAL NETWORKS FOR NLP. http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/.

[3] Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: VIPS: a Vision-based Page Segmentation Algorithm. Technical report, 2003.

[4] Chakrabarti, S.: Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In: *Proceedings of the 10th international conference on World Wide Web*, ACM, 2001, pp. 211–220.

[5] Dragúňová, M.: Vyhodnocovanie používateľského zážitku analýzou pohľadu a emócií. Bachelor thesis, Bratislava: FIIT STU, 2016.

[6] Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 2000, vol. 40, no. 10–12, pp. 1489 – 1506.

[7] Li, F.F., Karpathy, A., Johnson, J.: CS231n: Convolutional neural networks for visual recognition, 2015.

[8] Liu, Y., Wang, Q., Wang, Q. In: *A Heuristic Approach for Topical Information Extraction from News Pages*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 357–362.

[9] Mehta, R.R., Mitra, P., Karnick, H.: Extracting semantic structure of web documents using content and visual information. In: *Special interest tracks and posters of the 14th international conference on World Wide Web*, ACM, 2005, pp. 928–929.

[10] Ruihua Song, Haifeng Liu, J.R.W., Ma, W.Y.: Learning block importance models for web pages. In: *Proceedings of the 13th international conference on World Wide Web (WWW '04)*, New York, USA, 2004, pp. 203–211.

[11] Shen, C., Zhao, Q. In: *Webpage Saliency*. Springer International Publishing, Cham, 2014, pp. 33–46.

[2] https://www.tensorflow.org/