Feature Selection Using Visual Saliency for Content-Based Image Retrieval

Zbigniew Zdziarski[†] **and Rozenn Dahyot**[*]

School of Computer Science and Statistics
Trinity College Dublin,
Ireland

E-mail: [†]`zdziarze@tcd.ie` [*]`Rozenn.Dahyot@tcd.ie`

Abstract — **Saliency algorithms in content-based image retrieval are employed to retrieve the most important regions of an image with the idea that these regions hold the essence of representative information. Such regions are then typically analysed and described for future retrieval/classification tasks rather than the entire image itself - thus minimising computational resources required. We show that we can select a small number of features for indexing using a visual saliency measure without reducing the performance of classifiers trained to find objects.**

Keywords — **visual saliency, content-based image retrieval, image classification.**

I Introduction

Humans optimise the analysis of a scene by sifting through relevant and irrelevant information. Attention shifts from one 'important' region to another and it is during this time that the brain builds its representation [1]. This optimising action is the subject of research in visual saliency.

Most commonly, visual saliency is calculated by considering what stands out in an image, i.e. what can grab a viewer's attention. There are various definitions for 'what stands out' in images leading to different algorithms to calculate visual saliency. For example, Itti and Baldi use the term the 'surprise factor' [2], Schiele and Crowley in [3] and Gilles in [4] use 'rarity' or 'uniqueness', Julesz uses the term 'pop-out' [5]. But the common idea of visual saliency is to find the areas in an image/video which grab our attention on the global or local scale.

Content-based image retrieval (CBIR) is an area of research that aims at defining relevant visual features for efficient content retrieval in image databases. In this paper we investigate if visual saliency can help in selecting visual features for retrieval and consequently reduce the computation time and memory consumption needed for visual feature storage. With one parameter (threshold), we control the number of selected features in images used for retrieval. We show that even with a small number of features, the classifiers trained to find the objects of interest are still perform well. This result can be used to scale CBIR systems depending on the computational resources available on the device being used (e.g. tablets, mobiles, etc.).

II Previous Work

The notion of searching for attention grabbing regions has been transposed into various saliency models by computer scientists. Some models are biologically-based meaning that they are based on psychophysical and physiological evidence, while others are not. Many saliency models that have being proposed to the scientific community are founded on the biologically-based Koch and Ullman model [7]. It is based on psychological studies published in 1985 by Treisman and Gelade [8]. In the Koch and Ullman model, saliency is first calculated by extracting features (visual cues - such as colour, orientation, direction of movement, etc.) from an image. An activation map (an initial saliency map) is then calculated in parallel for each of these channels. Next, the activation maps are combined into a master saliency map. Finally, a ranking of salient regions is computed through the use of a Winner-Takes-All (WTA) network. The Koch and Ullman model has been used in many applications [9, 10, 11].

Several other saliency models have been proposed to the research community using a graph representation of images [2, 12, 13]. For instance, in the Graph-Based Visual Saliency (GBVS) algorithm [2], the edges of a graph are used to denote similarity between two nodes (pixels). Random walks are then performed on these nodes and the more a node is visited, the more salient it is deemed to be.

When using saliency in CBIR, a common approach among researchers is to find salient regions and then utilise feature point locating techniques on these regions only. This was for example the strategy used by Rutishauser et al. [14] and Walther et al. [15]. They chose to employ an additional segmentation step in their calculations by extracting objects from salient regions. Features were then located from these regions, i.e. from the extracted objects, rather than the regions defined by visual saliency algorithms. This segmentation step, however, has not always performed well in practice and still requires further work [16].

In this paper, we chose to use the GBVS algorithm for saliency calculations because it has been shown to be superior to the classic Koch and Ullman model in predicting human fixations [10]. The Speeded Up Robust Features (SURF) feature point algorithm developed by Bay in [17] is used to locate and describe feature points in this publication. This feature detection method was selected due to its robustness, memory efficiency in describing feature points and superior speed when compared to other popular feature detection methods such as the Scale Invariant Feature Transform (SIFT) [18] method [19].

No one has yet, to the best of our knowledge, analysed what the effect is of changing the saliency measure to decrease the size of extracted regions on classification results. The standard classification technique Support Vector Machines (SVM) is used to perform the classification on the selected SURF features using saliency information.

We present next in more detail our basic CBIR system and the experiments done to assess the usefulness of saliency for this application.

III CBIR System Design

A data set of three image classes were collected: horses, yellow flowers, and faces. The horses data set was obtained from the INRIA Horses V1.03 data set [20]. The yellow flowers data set was compiled from images from the Visual Geometry Group's (The University of Oxford) set [21]. The faces data set was a combination of the 'face94' Essex face database [22], the MIT CBCL Face data set [23] and the Caltech101 data set [24]. All faces chosen from the data set were of head-on shots and each data set was used separately. The three image classes were selected because they are a good representation of objects with differing colour, texture and context.

		Correctly classified	Incorrectly classified
Horses	$\mathbf{C}$	84.5%	15.5%
	$\bar{\mathbf{C}}$	98.7%	1.3%
Flowers	$\mathbf{C}$	80.1%	19.9%
	$\bar{\mathbf{C}}$	97.8%	2.2%
Faces	$\mathbf{C}$	90.3%	9.7%
	$\bar{\mathbf{C}}$	99.9%	0.1%

Table 1: Percentage of training features correctly and incorrectly classified by the SVM.

The first step in the experiment was to train a two-class SVM. For each data set, 100 positive images were cropped to only include the object of interest: horse, yellow flower or face. These images and 100 negative images (taken from the negative images folder of the INRIA Horse V1.03 data set) were then scanned for SURF feature points. All the calculated feature points were then fed into the SVM. Classification results of the SVM for the training features are shown in Table 1, with $\mathbf{C}$ representing feature points correctly classified in the positive group and $\bar{\mathbf{C}}$ feature points correctly classified in the negative group.

The GBVS algorithm gives a saliency value between 0 and 1 for each pixel from an image, where 0 signifies no saliency, and 1 indicates the highest saliency value possible. Entire regions can have saliency values of 1 and the distribution of saliency values can vary depending on each image.

A different subset of images to the training set was used for testing. For each object of interest (horse, flower, face), 40 positive ($\mathbf{P}$) and 40 negative ($\mathbf{N}$) images were tested. The objects in the positive image group cover approximately 25% to 35% of the total size of the image (contrary to the images used to trained the SVM classifiers, the test images of the objects of interest were not cropped). Some examples of the images in $\mathbf{P}$ can be seen in Figure 3.

SURF features are computed for these test images, however only the features associated with a saliency above a threshold $\mathbf{T}$ are fed into the classifier to try to find the images with the object of interest.

The following saliency thresholds were used in our experiment: $\mathbf{T} = 0$ (no saliency information used), 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. The higher the value of **T**, the fewer number of feature points fed into the SVM. Figure 1 shows the training and testing flowchart of the experiment.

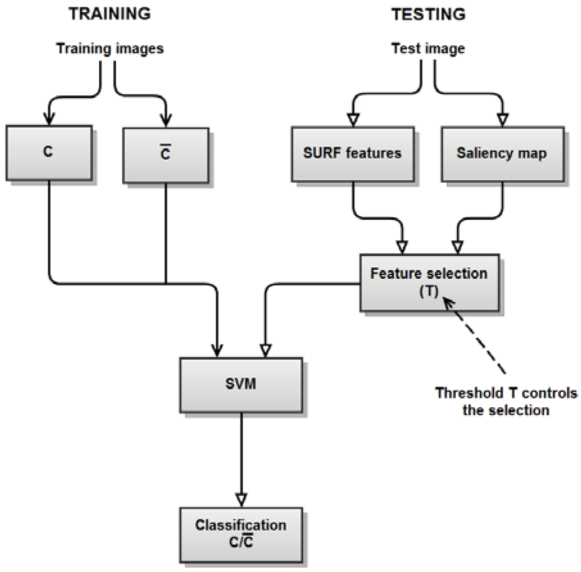


Fig. 1: Flowchart of the training and testing phases of the experiment

IV Performance Using Visual Saliency

The test set of positive images for each object of interest were treated as one large image by collecting all the feature points found in all images into one pool. The pool of features from the positive images were run separately in the SVM to the pool of features from the negative images. For each threshold we compute the following proportion:

$$p_p = \frac{a_p}{n_p}\ , \qquad p_n = \frac{a_n}{n_n} \tag{1}$$

where a_p is the number of feature points in **P** classified as **C**, a_n is number of feature points in **N** classified as **C**, n_p is the total number of feature points detected in **P** and n_n is the total number of feature points detected in **N**. The confusion matrix corresponds to:

	C	$\bar{\mathbf{C}}$
P	p_p	1-p_p
N	1-p_n	p_n

A standard error can be computed to measure the uncertainty associated with the proportions p_p and p_n such that:

$$SE(p) = \sqrt{p(1-p)/n} \tag{2}$$

where p is p_p (resp. p_n) and n is n_p (resp. n_n). The proportion p_p and $1 - p_n$ for each data set was computed for each threshold value **T** with its corresponding standard error appearing as error bars. If selecting features using saliency can reduce computation time, we want to check that the retrieval result will not deteriorate as less features are considered for classification.

a) Classification results

Figures 2 (a), (b) and (c) show these plots for the horses, yellow flowers and faces data sets respectively. The positive classifications p_p improves as the saliency threshold increases for all objects (with $0 \leq \mathbf{T} \leq 0.8$). The improvement with saliency (**T**= 0.8) versus without saliency (**T**=0) is of 10% for Horses, 12% for Flowers, and 12% for Faces. This seems to indicate that the feature points that are left out for classification as **T** increases are more often from $\bar{\mathbf{C}}$ (background) than the class of interest **C** (horse, flower, face). On the other hand, the (false positive) proportion $1-p_n$ measuring the proportion of points classified in **C** in the negative set of images (where no object of interest appears) remains more or less constant whatever the saliency level chosen.

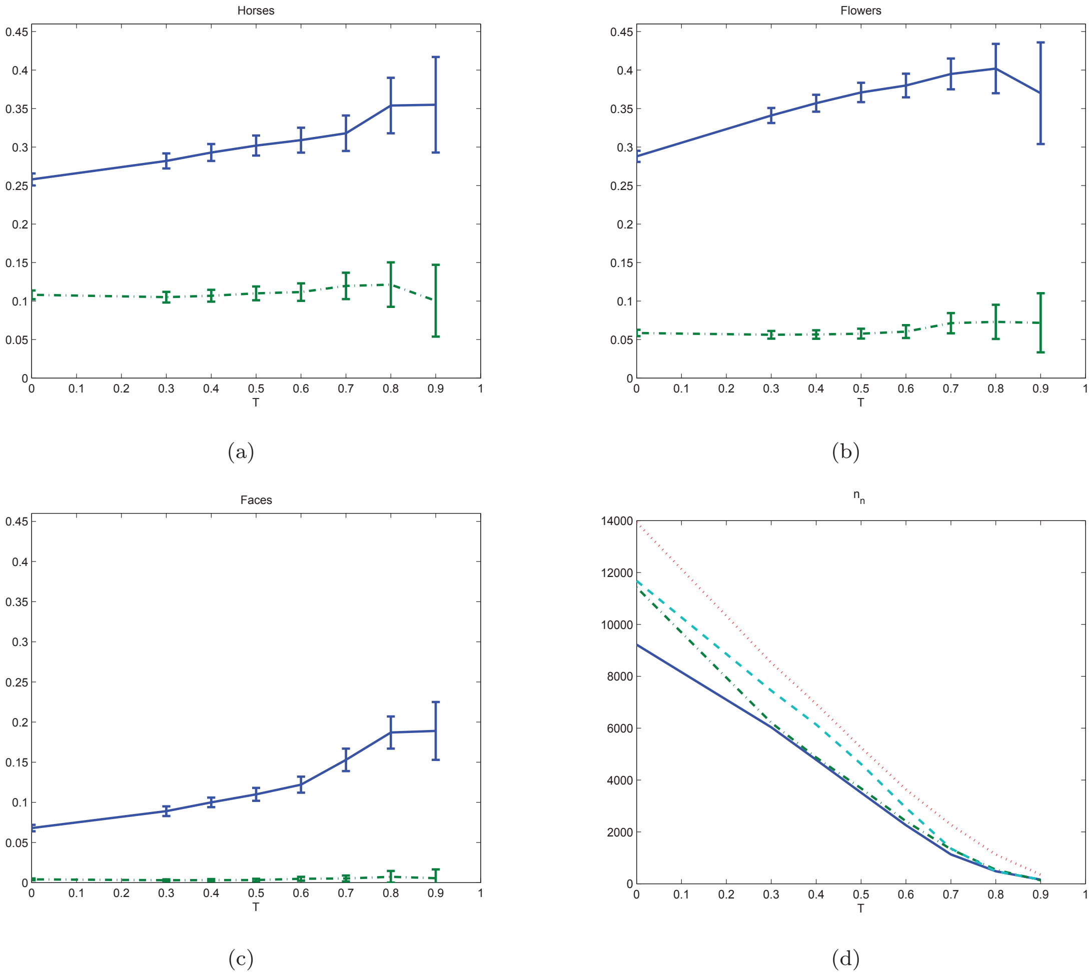
This first result indicates that selecting features using saliency will not deteriorate classification performances but can in fact improve them.

As the saliency threshold increases, the standard errors for the proportions also increase. This is due to the fact that the number of selected features, n_p and n_n, decrease as **T** increases (see equation 2). Although p_p generally improves as the threshold increases, the certainty about how much the proportion actually improves is reduced. It can also be noticed that the standard errors for faces are smaller at each value of **T** compared to the other two classes. The reason for this is that more feature points were being detected in this class of images as opposed to flower and horse images. Most of the face images were taken with relatively ‘busy’ backgrounds that foster bountiful amounts of feature point detection. The difference in the number of feature points detected in the three classes can also explain the low p_p values obtained for the faces data set. More background points will bring down the value of p_p.

Figures 3 shows some results of the feature classification on images of the positive classes for two saliency thresholds. Although the classifier (SVM) used is not optimal (some points on the object are misclassified as $\bar{\mathbf{C}}$ while some points on the background are classifed as **C**), the objects of interest are well found by many feature points even as the saliency threshold increases.

b) Disk Space Usage Results and Discussion

Another aspect of this experiment was to see what the effect of changing the value of **T** would have on disk space usage and classification time. Figure 4 (a) shows a plot of the memory usage required



(a)

(b)

(c)

(d)

Fig. 2: (a): proportion of correct classification for the horses data set p_p (solid blue line) and $1 - p_n$ (green dashdot line) w.r.t. T; (b) proportion of correct classification for the flowers data set p_p (solid blue line) and $1 - p_n$ (green dashdot line) w.r.t. T; (c) proportion of correct classification for the faces data set p_p (solid blue line) and $1 - p_n$ (green dashdot line) w.r.t. T; (d) total no. of feature points n_p for horses (solid blue line), flowers (green dashdot line) and faces (dotted red line) and n_n (aqua dashed line).

for storing feature points in memory for the three classes of images. A SURF feature point is a 64 element vector of floating point numbers. Since the size of doubles stored in memory is machine dependent, a single unit of memory usage was used for a double to abstract over this machine dependence (1 double = 1 unit, 1 SURF feature point = 64 units).

Figure 4 (a) depicts a clear downward trend for memory usage for all three classes. This trend is linear until $\mathbf{T} = 0.7$ when the rate of change decreases. At $\mathbf{T} = 0.5$, for example, a 60% decrease in required memory usage is obtained, while a 75% result is obtained at threshold value 0.6.

Figure 4 (b) shows the computation time that was clocked for classifying the images in the three image classes. These results were obtained on a 2.67 GHz Intel Core2 Quad CPU running Windows 7 with 4 GB of RAM. The classifying application (libsvm [25]) was used in Matlab.

The computation time graph is very similar to the memory usage graph of Figure 4 (a). One would expect this as the classification computation time is directly proportional to the number of features stored in memory, assuming all the feature points can be easily stored in RAM. An approximate speed-up of 60% and 75% was recorded for the **T** values of 0.5 and 0.6 respectively, which is the exact same level of improvement recorded for memory usage.

V Conclusion

This paper showed that visual saliency can be used to help in efficiently selecting visual features in a retrieval system. Selection of features using visual saliency reduces computation time and memory consumption needed for visual feature storage. The saliency threshold can be tuned efficiently to get the best retrieval performance for both accuracy and speed, and it can be used to scale the

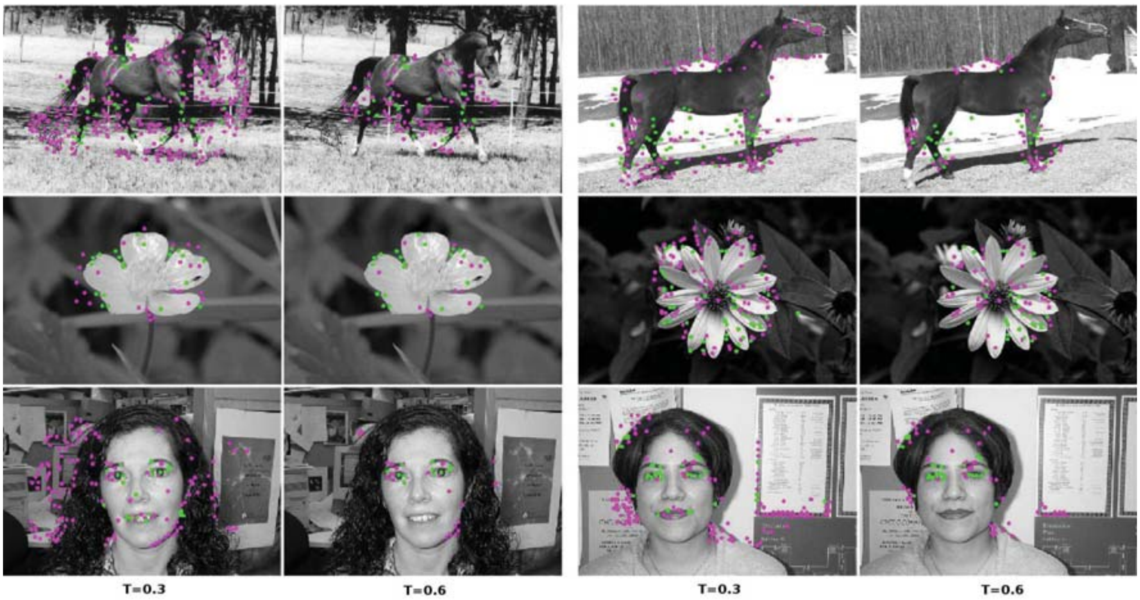


Fig. 3: Example results for $\mathbf{T} = 0.3$ and 0.6. From top to bottom: horses, flowers and faces. Green dots indicate feature points classified in the image class ($\mathbf{C}$), purple dots indicate feature points classified outside of it ($\bar{\mathbf{C}}$).

system to different devices.

References

[1] J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it. *Nature Reviews Neuroscience*, 5:495–501, 2004.

[2] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 19:545–552, 2007.

[3] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. *Europ. Conf. on Comp. Vision (ECCV)*, 1064:610–619, 1996.

[4] S. Gilles. *Robust Description and Matching of Images.* PhD thesis, Uni. of Oxford, 1998.

[5] B. Julesz. *Dialogues on Perception.* MIT Press, 1995.

[6] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.

[7] C. Koch and S. Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurology*, 4:219–227, 1985.

[8] A. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[9] R. Milanese, H. Wechsler, S. Gil, J. Bost, and T. Pun. Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. *Computer Vision and Pattern Recognition*, pages 781–785, 1994.

[10] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[11] A. Maki, P. Nordlund, and J.-O. Eklundh. A computational model of depth-based attention. *Int'l Conf. on Pattern Recognition*, 4:734–738, 1996.

[12] L. de Fontoura Costa. Visual saliency and attention as random walks on complex networks. arXiv preprint, 2006.

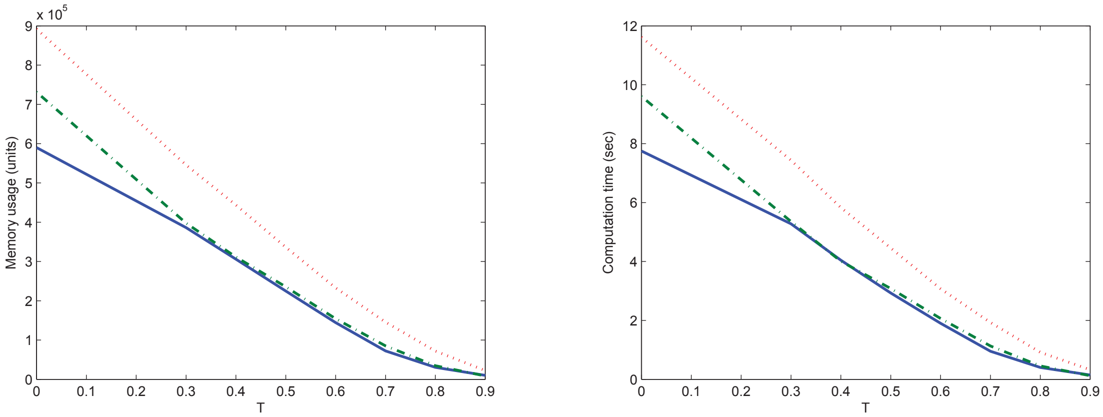
[13] V. Gopalakrishnan, Y. Hu, and D. Rajan. Random walks on graphs to model saliency in images. *Computer Vision and Pattern Recognition*, pages 1698–1705, 2009.

[14] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? *Proc. of Computer Vision and Pattern Recognition*, 2:37–44, 2004.

[15] D. Walther. Modeling attention to salient proto-objects. *Neural Networks*, 19(9):1395–1407, 2006.

[16] T. Kadir and M. Brady. Saliency, scale and image description. *Int'l Journal of Comp. Vision*, 45(2):83–105, 2000.

[17] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *European Conf. on Computer Vision*, 1:404–417, 2006.



(a)

(b)

Fig. 4: (a) Memory usage for positive images w.r.t T for horses (solid blue line), flowers (green dashdot line) and faces (red dotted line); (b) Classification computation time for positive images w.r.t. T for horses (solid blue line), flowers (green dashdot line) and faces (red dotted line).

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision*, 60(2):91–110, 2004.

[19] L. Juan and O. Gwun. A comparison of sift, pca-sift and surf. *Int'l Journal of Image Processing*, 3(4):143–152, 2009.

[20] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. *Computer Vision and Pattern Recognition*, 2:90–96, 2004.

[21] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. *Int'l Conf. on Comp. Vision*, 2:1447–1454, 2006.

[22] Available at: `http://cswww.essex.ac.uk/mv/allfaces/faces94.zip`, retrieved 15/3/2012 at 10:40.

[23] B. Weyrauch, J. Huang, B. Heisele, and V. Blanz. Component-based face recognition with 3d morphable models. *First IEEE Workshop on face processing in video*, 2004.

[24] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *CVPR Workshop on Generative-Model Based Vision*, 12:178, 2004.

[25] C.-C. Chang and C.-J. Lin. Libsvm – a library for support vector machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`, retrieved 23/3/2012 at 11:00.