EECS 4422/5323 Assignment 2

Due: 15 November, 2019

Saliency-based Thumbnail Generation

This assignment involves automated generation of more informative image thumbnails based on saliency maps. You will be provided with a dataset of 100 images and the corresponding saliency maps for those images from six different saliency models: Attention by Information Maximization (AIM) [1], Boolean Map Saliency (BMS) [2], DeepGaze II (DGII) [3], the Itti-Koch-Niebur Saliency Model (IKN) [4], Image Signature (IMSIG) [5], and Saliency using Generative Adversarial Networks (SalGAN) [6]. You will also be provided with two types of human-generated ground truth data: object masks from the Salient Objects in Clutter (SOC) dataset [7], and "human saliency maps" from the Saliency in Context (SALICON) dataset [8].

All of these files are available to you in a zip file called `Assignment_2_files.zip` available for download on the course website. Inside the zip folder you will find the following contents:

- `images`: A folder containing the 100 input images
- `human_maps`: A folder containing the SALICON maps
- `masks`: A folder containing the SOC object masks

NOTE: All of your code should be included in a file with this exact filename: `Assignment_Two.py`. This file should be importable via the command `import Assignment_Two` (you should test this yourself before submitting). All functions specified below should be included in this file *in the exact format specified*.

Part 1: Generating Thumbnail ROIs [60 Marks]

Your first step in the assignment is to generate regions of interest (ROIs) which could be resized to serve as the image thumbnail. The goal is to locate a subset of the image which is as small as possible (to mimize the degree of downsampling required when converting to a thumbnail) while retaining maximum salient information.

You will do this by writing a function with the following format:

```
roi_coords = generate_roi(salmap, thresh, alpha, beta)
```

with inputs and outputs as follows:

- `salmap`: A saliency map corresponding to the input image. This will be a grayscale image, but may have one or three channels.
- `thresh`: A saliency threshold between 0 and 100 which denotes the percentage of pixels in the saliency map to be marked as above threshold (in rank order)
- `alpha`: A scalar parameter corresponding to α in Equation 1
- `beta`: A scalar parameter corresponding to β in Equation 1
- `roi_coords`: A set of coordinates denoting a subset ROI of the input image, as described below, in the format of a 4-tuple (x, y, w, h)
  - x The x-coordinate of the top left corner of the box
  - y The y-coordinate of the top left corner of the box
  - w The width of the box
  - h The height of the box

The ROI produced by this function should have the same aspect ratio as the original input image, and correspond to the image subset which maximizes the ROI score (R_s) as calculated in the following equation:

$$R_s = \alpha \left(\sum_{P \in \mathbf{R}} S_P \right) - \beta N \tag{1}$$

where $\mathbf{R}$ is the set of pixels forming the ROI, P is a pixel in the ROI, S_P is the binarized saliency score of the pixel P (1 if the saliency of the pixel exceeds the threshold, 0 otherwise), and N is the total number of pixels in $\mathbf{R}$.

Note that this equation penalizes large ROIs while simultaneously encouraging the retention of pixels with high saliency values.

Clarification, 7 November: When salient pixels are tied in value, compute the value which retains at least the percentage of pixels specified by threshold. Any pixels with saliency score equal to this value should be retained.

Clarification, 7 November: If you use any heuristics in your search for the optimal ROI in order to increase the speed of finding it, please briefly explain the nature of the heuristic(s) in comments above your function.

Hint: In order to efficiently compute ROI scores such that the maximum ROI can be found, it is recommended that you utilize an integral image datat structure.

Part 2: Evaluating Predicted ROIs with Object Masks [45 Marks]

In this part of the assignment, you are to evaluate the set of saliency models with provided output for their ability to predict image ROIs which contain objects identified by human annotators as salient. To do this, you must implement a function with the following format:

`[precision, recall] = mask_score_roi(mask, roi)`

- `mask`: A ground-truth object mask as provided to you.
- `roi`: A 4-tuple denoting a target box, as produced by the function `generate_roi`
- `precision`: The precision of the input ROI, computed as described below
- `recall`: The recall of the input ROI, computed as described below

For the purposes of this computation, a true positive is when the ROI includes a pixel which is part of the ground-truth object mask, a false negative is when a pixel part of the ground-truth object mask is not included in the ROI, and a false positive is when the ROI includes a pixel which is outside of a minimal bounding box around the ground-truth object mask.

Using this scoring function, produce a PDF report, `A2_Mask_Eval.pdf`, included with your code submission which contains the following content:

- Produce a precision-recall (PR) curve showing the results for each tested saliency model varying threshold from 10 to 90 in steps of 10 for the following ROI generating parameter settings (this should be a separate plot for each set of parameters, with each plot showing curves for all six models).
  - $\alpha = 0.8, \beta = 0.3$
  - $\alpha = 1.0, \beta = 0.1$
  - $\alpha = 0.6, \beta = 0.4$
- For each setting given above, compute the area under the PR curve for each model and report which model achieves the highest area-under-the-curve (AUC) score.
- [*5323 only*] Provide a brief (one paragraph) discussion of your results, commenting on how the parameter settings affect the precision and recall of the models, and any thoughts you might have on why a given model performed well or poorly.

All code used to produce the figures and calculate the area-under-the-curve should also be included in your zip file, though the formatting is up to you.

Clarification, 7 November: The PR curves produced for this part should include the aggregate performance of the saliency models over all 100 images provided.

Part 3: Evaluating Predicted ROIs with "Human Maps" [45 Marks]

In this part of the assignment, you are to evaluate the set of saliency models with provided output for their ability to predict image ROIs based on their correspondence to the parts of the image attended to by human observers as measured by mouse clicks over a degraded image (see [8] for more details on how the ground truth data was gathered). To do this, you must implement a function with the following format:

`[score, missed] = salience_score_roi(map, roi)`

- `map`: A ground-truth "human saliency map" as provided to you
- `roi`: A 4-tuple denoting a target box, as produced by the function `generate_roi`
- `score`: A score value computed as the average pixel value of the human saliency map contained within the ROI
- `missed`: The total value of all pixels in the human saliency map which are not contained within the ROI

Using this scoring function, produce a PDF report, `A2_Human_Eval.pdf`, included with your code submission which contains the following content:

- A plot each of `score` versus threshold and `missed` versus threshold (varying from 10 to 90 in steps of 10, and with the results for all six models on each plot) for each of the following parameter settings:
  - $\alpha = 0.8, \beta = 0.3$
  - $\alpha = 1.0, \beta = 0.1$
  - $\alpha = 0.6, \beta = 0.4$
- Provide a brief (one paragraph) discussion of how the pattern of model results from this test compare to the pattern of model results from the evaluation based on object masks.
- [*5323 only*] Provide a brief (one paragraph) discussion of your results, commenting on which model's pattern of behaviour from this evaluation indicates the best performance for this task.

All code used to produce the figures in your PDF should also be included in your zip file, though the formatting is up to you.

Clarification, 7 November: The plots produced for this part should include the aggregate performance of the saliency models over all 100 images provided.

References

[1] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 18, pp. 155–162, 2006.

[2] J. Zhang and S. Stan, "Exploiting surroundedness for saliency detection: A Boolean map approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 5, no. 38, pp. 889–902, 2016.

[3] M. Kümmerer, T. S. A. Wallis, and M. Bethge, "DeepGaze II: reading fixations from deep features trained on object recognition," *arXiv*, vol. abs/1610.01563, 2016.

[4] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 20, pp. 1254–1259, 1998.

[5] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, pp. 194–201, 2012.

[6] J. Pan, C. Canton-Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giró i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv*, vol. abs/1701.01081, 2017.

[7] D.-P. Fan, M.-M. Cheng, J.-J. Liu, S.-H. Gao, Q. Hou, and A. Borji, "Salient objects in clutter: Bringing salient object detection to the foreground," in *European Conference on Computer Vision (ECCV)*, Springer, 2018.

[8] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "SALICON: Saliency in context," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.