Large-scale Scene Understanding Challenge: 
Eye Tracking Saliency Estimation

Yinda Zhang, Fisher Yu, Shuran Song, Pingmei Xu, Ari Seff, Jianxiong Xiao
Princeton University

1 Task description

The objective of eye tracking saliency challenge is to generate a saliency map (Fig. 1(c)), which can predict the ground truth saliency map and fixation points (Fig. 1(b)).

2 Data

We provide are 6000 images for training, 926 for validation, and 2000 for testing. Please download zip files for image, fixation, and saliency map and unzip them in to a same folder, e.g. Root. The raw images are collected from SUN database [2], and the eye tracking saliency ground truth are collected from crowd sourcing platform (Amazon Mechanic Turk) using the method described in [3]. Each image has been viewed by 3-10 subjects.

The training set and validation set, provided with ground truth, contains the following data field:

– **image**: The name of the image.

  The image can be found at “Root/images/image.jpg”. The ground truth saliency map can be found at “Root/saliency/image.mat”. The ground truth binary fixation map can be found at “Root/fixation/image.mat”.

– **resolution**: The image resolution [height, width].

– **scenecategory**: The scene type of the image. This is an additional information to encourage scene-related algorithms. Whether to use the scene type or not is a free option, and we will compare algorithms with and without using scene type separately.

– **gaze**: The ground truth gaze data from subjects. Each structure corresponds to one subject, and there are no less than 3 subjects per image. Each gaze structure contains:

  • **location**: the image location of each gaze point, [x,y].
  • **timestamp**: the time stamp (millisecond) of each gaze point.
  • **fixation**: the fixation points estimated by mean-shift, [x,y].

The testing data contains only **image**, **scenecategory**, and **resolution** fields.

People may choose whether to use **scenecategory** for prediction freely but are required to report this in the submission.
Fig. 1: **Task.** We collect fixation points (b) and saliency map (c) from the crowd sourcing platform using the method described in [3]. The task is to generate a saliency map from the input image, which can predict the ground truth fixation points and saliency map.

### 3 Evaluation metrics

We adopt most of the standard metrics provided in MIT saliency benchmark[1] defined on both saliency map and fixation points. Specifically, we will evaluate the following metrics:

- Similarity
- CC
- AUC_Judd
- AUC_Borji
- sAUC (AUC_shuffled)

Please refer to the MIT saliency benchmark for more details.

### 4 Toolkit

- **demo.m.** General pipeline about how to use the toolkit.
- **GlobalParameters.m.** Define global parameters. You should set up “ROOT_DIR” to the root folder of the data.
- **predictFunc.** An example showing what to output for a prediction function.
- **evaluationFunc.** The evaluation function we will call on the server. It will take prediction, ground truth, and a metric type as input, and output the performance under the metric. The metric name can be one of the “similarity”, “CC”, “AUC_Judd”, “AUC_Borji”, and “AUC_shuffled”.
- **makeFixationMap.** Convert fixation points to binary map.
- **code_forMetrics.** Codes from MIT saliency benchmark.

### 5 What to submit

Participants are supposed to run their algorithm on testing set and organize the result in the format exactly the same as the output of the **predictFunc.** The result is an array of cells. Each cell contains the predicted saliency map for the corresponding images in data, which should be validation or testing.
References