Cognitive Computer Vision: Eye fixation prediction project

1 Introduction

Figure 1 shows an example of an input image and an eye fixation map that has been collected with the help of an eye tracker device. Many observers are shown the image, and their eye movements are tracked. The fixation map is produced by averaging over the fixations of many observers. The objective of this project is to implement a machine learning based system to predict human eye fixations.

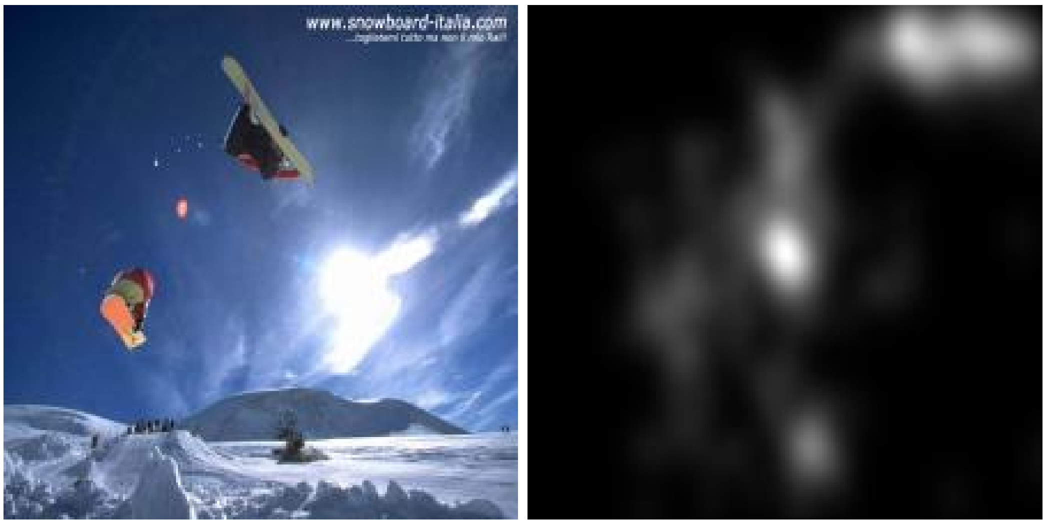

Figure 1: Example of an input image (left) and the corresponding eye fixation map (right).

2 Project requirements and instructions

To successfully pass the course project, you must

1. implement and train a machine learning based eye fixation prediction system using Tensorflow,
2. evaluate the system's performance,
3. submit your source code and results, and
4. give a presentation on your system, the training procedure, and the results you obtained.

The last date for submitting your code and results is June 12th 2018. Observe these other instructions as well.

- You can complete the project on your own or in a group of 2 students. Identify clearly the group members in all submitted work.

- You are free to use any neural network architecture you want to implement the system. A few papers to look into for some ideas are [2, 4, 3].
- You may use any dataset except the given test set to train your system.
- You may use transfer learning, i.e., take an existing network with already learned weights, and use it as a basis for your system.
- It is encouraged that you look at some existing systems and take inspiration from them.
- However, do *not* take an existing implementation, i.e., code written by another person, of another system and submit it as your own work. Doing so will result in failing the project.

3 Datasets

You can download the dataset via the course Moodle page. The dataset is split into three sets:

- Training, with 1200 images and corresponding fixation maps,
- Validation, with 400 images and corresponding fixation maps, and
- Testing, with 400 images.

The images come from several different categories such as "art", "action", or "cartoon". Each of the images is of size 224-by-224, with 3 channels[1]. Each of the fixation maps is of size 224-by-224, with one channel. Figure 1 shows an example pair of an input image and a fixation map. Note that for the testing dataset, the fixation maps are not provided. You should tune the performance of your system using the training and validation datasets, and verify that it gives reasonable results on the test dataset, e.g., through visually inspecting the results. The course instructors have the fixation maps for the testing dataset and will evaluate the submitted systems to rank them.

4 Evaluation

To evaluate how good the predicted fixation maps are, we will use the Kullback-Leibler divergence (KLD). KLD is a quantification of the difference between two probability mass functions: a low KLD indicates the distributions are similar, and vice versa for high KLD. Let P and G denote the predicted and ground truth fixation maps, respectively. The KLD is a non-symmetric measure of the information lost when P is used to estimate G. Additional information about KLD and alternative evaluation metrics can be found in [1].

Let P_i denote the value of the ith pixel in the predicted fixation map, and let G_i denote the value of the ith pixel in the ground truth fixation map. Then, KLD is calculated as

$$KLD(P, G) = \sum_i Q_i \log\left(\frac{Q_i}{P_i}\right).$$

You will submit a set of predicted fixation maps for the images in the test set. The course instructors will then evaluate them against the ground truth using KLD. The submitted predicted fixation maps, and the ground truth fixation maps, will be interpreted as probability mass functions. You can experiment with the KLD yourself using the function provided in `kld.py`, which you will find in Moodle.

[1]Some of the images are still effectively grayscale, as the R, G, B channel contents are identical.

The final evaluation criterion is the average KLD over all the images in the test set. You should not use KLD as the loss function for your network, but it can be helpful to understand how the end result will finally be evaluated. For example, you can calculate the average KLD over the validation set to figure out how well your system is doing.

5 How to submit the results?

There are 400 testing images in the test dataset, named `1601.jpg` to `2000.jpg`. You must submit a predicted fixation map output by your system for each of the testing images.

Naming the files. For an image named, for example, `1689.jpg`, save the predicted fixation map output by your method as `1689_prediction.jpg`. Repeat for every image in the test set.

Output type of files. The predicted fixation maps should be stored as 224 by 224 images with a single channel (grayscale), using JPEG encoding.

Submitting the results. Remember to convert to `uint8` type for the images, and scale them to the appropriate range $[0, 255]$ before doing the type conversion.

Archiving the fixation map files. Place all the predicted fixation maps inside an archive with the name `results_student_name1_student_name2.zip`, if your group consists of members "Student Name1" and "Student Name2".

Source code. Create another archive with the name `source_student_name1_student_name2.zip` for your source code `.py` files.

File submission. Upload the two `.zip` files through Moodle before the deadline. If you work in a group of two, only one student needs to upload the files.

References

[1] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *arXiv preprint arXiv:1604.03605*, 2016.

[2] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*, 2016.

[3] S. S. S. Kruthiventi, K. Ayush, and R. V. Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *IEEE Transactions on Image Processing*, 26(9):4446–4456, 2017.

[4] Matthias Kümmerer and Lucas Theis and Matthias Bethge. Deep Gaze I: Boosting Saliency Prediction with Feature Maps Trained on ImageNet. In *ICLR 2015 Workshop*, 2015.