Computer Vision 2: Eye fixation prediction project

Summer semester 2020

1 Introduction

Figure 1 shows an example of an input image and an eye fixation or saliency map. The eye fixation map is collected with the help of an eye tracker device. Many observers are shown the image, and their eye movements are tracked. The eye fixation map is produced by averaging over the fixations of many observers and smoothing with a Gaussian kernel. The objective of this project is to implement a machine learning based system for eye fixation prediction. This system is able to produce maps such as shown on the right given an input image.

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 eye fixation prediction system using PyTorch, and
2. submit a written description of your system, along with your source code and results.

- **The deadline for submitting your code and results is June 12th 2020, at 23:59.**
- You may use any neural network architecture you want to implement the system. The papers by Pan et al. (2017); Kummerer et al. (2017); Cornia et al. (2016); Cornia et al. (2018); Kruthiventi et al. (2017) can provide useful ideas.
- 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.
- You may reuse any code and examples from the exercises in your project.
- Do not take an existing *implementation*, i.e., code written by another person, and submit it as your own work. Doing so will result in failing the project.

3 Dataset

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

- Training, with 3006 images and eye fixation maps,
- Validation, with 1128 images and eye fixation maps, and
- Testing, with 1032 images.

Each image is of size 224-by-224, with 3 channels. Each fixation map is of size 224-by-224, with one channel. Figure 1 shows an example input image and fixation map. Note that for the testing dataset, the fixation maps are not provided. You can 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 eye fixation maps for the testing dataset and will evaluate the submitted systems to rank them.

All of the `.zip` files containing data should be extracted to the same folder. Five text files `train_images.txt`, `train_fixations.txt`, `val_images.txt`, `val_fixations.txt`, and `test_images.txt` are also provided.

If the dataset is saved to a folder with the name `/home/username/cv2_project_data/`, then this folder name can be concatenated with any line of these text files to obtain a valid name for an image or eye fixation map. For example, the first line of `train_images.txt` contains `images/train/image-0.png`. Then `/home/username/cv2_project_data/images/train/image-0.png` is a valid image in the training set.

Lines in `train_images.txt` and `train_fixations.txt` are paired: the ith row in both files points to a related pair of input image and eye fixation map. The same is true for `val_images.txt` and `val_fixations.txt`.

4 How to submit the results?

The testing images in the test dataset are named `image-4134.png` to `image-5167.png`. You must submit a predicted eye fixation map output by your system for each of the testing images.

Naming the files. For an image named, for example, `image-4134.png`, save the predicted eye fixation map output by your method as `prediction-4134.png`. Repeat for every image in the test set.

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

Archiving the eye fixation map files. Place all the predicted eye fixation maps inside an archive with the name `lastname_firstname_predictions.zip`, if your name is “Firstname Lastname”.

Source code. Create an archive with the name `lastname_firstname_source.zip` for your source code `.py` files. **Do not** include data (images, network weights, etc.) in the zip file.

System summary. Prepare a written document (maximum length 2 A4 pages) which answers the following questions:

- What existing work is the system based on? Provide a reference to a paper.
- If you made modifications compared to the reference work, please provide a brief summary of them (a few sentences).
- Which loss function was used? What was the best loss value achieved on the validation data?
- How many epochs was the network trained for?
- Which hyperparameters were used (learning rate, dropout, data augmentation, other training procedures, ...)? You may provide figures or graphs if you want.

Save the file as PDF and name it `lastname_firstname_summary.pdf`.

File submission. Upload the two `.zip` files and the summary `.pdf` file through Moodle before the deadline.

References

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.

Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, October 2018. ISSN 1057-7149. doi: 10.1109/TIP.2018.2851672.

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.

Matthias Kummerer, Thomas S. A. Wallis, Leon A. Gatys, and Matthias Bethge. Understanding low- and high-level contributions to fixation prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Junting Pan, Cristian Canton, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol, and Xavier and Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. In *arXiv*, January 2017.