# Computer Vision 2: Eye fixation prediction project
# Deadline: 14.06.2019

## 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.

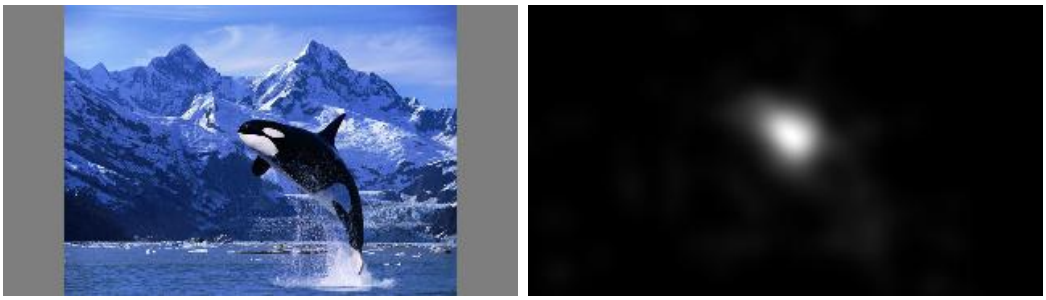

Figure 1: Example of an input image (left) and the corresponding eye fixation map (right).

## 2 Project requirements and instructions

To successfully pass the course project, you must

1. implement and train an eye fixation prediction system using Tensorflow,
2. submit your source code and results, and
3. give a presentation on your system and the results.

- **The deadline for submitting your code and results is June 14th 2019, at 23:59.**
- **You must work as a pair, in a group of 2 students.** Identify clearly both students in all submitted work.
- You may use any neural network architecture you want to implement the system. The papers by 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.

- 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 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 image is of size 180-by-320, with 3 channels[1]. Each fixation map is of size 180-by-320, 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 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 $G$.

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

$$KLD(G, P) = \sum_i G_i \log \left( \epsilon + \frac{G_i}{P_i + \epsilon} \right),$$

where $\epsilon > 0$ is a small regularization constant added to the usual definition to handle the numerically problematic cases.

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 as follows:

1. Apply a softmax function to each prediction $P$ and ground truth image $G$ to normalize them, that is for every pixel $i$,
$$P_i^n = \frac{\exp(P_i)}{\sum_j \exp(P_j)},$$
and similarly to get $G^n$. Then calculate $KLD(G^n, P^n)$.
2. Average the KLD's over all images.

The evaluation will use the function `kld.py`, which you will find in Moodle.

# 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.

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

**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 180 by 320 images with a single channel (grayscale), using JPEG encoding.

**Submitting the results.** Convert images to `uint8` type before saving them. Remember to scale appropriately.

For data type conversions, see `https://www.tensorflow.org/api_docs/python/tf/image/convert_image_dtype` (Tensorflow) or `http://scikit-image.org/docs/dev/user_guide/data_types.html` (Python/skimage).

**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. Only one group member needs to upload the files – but remember to write the names of all group members in your submitted files!

# 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.