

LABORATORY WORK 4 – PREDICTION OF VISUAL SALIENCY IN VIDEO SEQUENCES BY DEEP CNN

The visual saliency concept refers to the visual property that some objects around us are more perceived than others, they "capture our attention."

It is very important for a complex biological system (humans, animals) to be able to quickly detect potential prey, predators, or others in a world of visual information. Primates or other animals that filter visual information in order to focus their unconscious process of recognition of objects on a limited number of objects called "salient".

Many computer vision works are inspired by this visual property to focus their calculations on the salient areas. In the following figure, it is easy to see that for a saliency prediction in natural video an object in motion, here the two people running, attract human gaze.



The left figure shows the original frame. The right figure shows a "heat map" of the saliency values, called "saliency map" automatically calculated using an algorithm exploiting the properties of human visual system.

In this example, it is visually easy to conclude that the automatic extraction algorithm of salience shows very high performance. Nowadays, prediction of saliency in images and video by machine learning methods is an intensive research subject. Deep Convolutional Neural Networks (Deep CNNs) have proven to be a powerful mean for prediction of salient areas in stills. By adding residual motion information to the input of Deep CNNs, the architecture of the network ChaboNet developed in the LABRI to predict saliency in natural video is summarized in next figure.

The learning of Deep CNN parameters is frequently done with the technique of Stochastic Gradient Descent (SGD). Generally, for a simple supervised learning the SGD algorithm is as follows:

repeat until convergence

$$\begin{cases} V_{i+1} = \epsilon \cdot V_i - \gamma \cdot \alpha \cdot W_i - \alpha \cdot 1/N \cdot \sum \frac{\partial}{\partial W_i} f(W_i) \\ W_{i+1} = W_i + V_{i+1} \end{cases}$$

where W_i are network parameters of each convolutional layer at the iteration i , W_i is randomly initialized from a gaussian distribution, α is the learning rate, ϵ is a momentum coefficient, γ is the weight decay the sum is taken over a batch of size N and f is the loss function to minimise. V_i presents the previous weight update that is initialized at zero. The main SGD problem is that it converges to a local optimum even if α is constant. However, it is still the best learning algorithm when the training set is large.

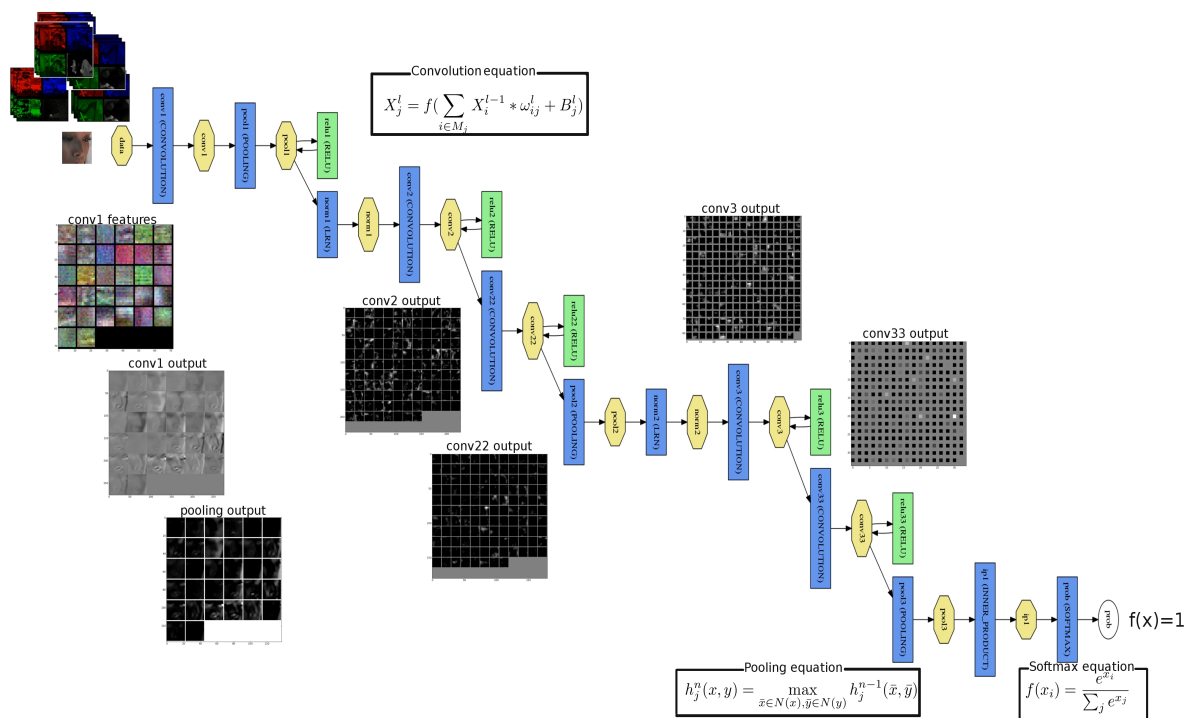


Figure : “ChaboNet” deep CNN to saliency prediction in natural video.

The model of “ChaboNet” network is already defined and trained. The training step needs an interest computational time. To simplify, you will just use it to start the prediction of saliency in some videos. The first point of interest: **how to use a trained model to create a dense saliency map**. It is visually easy to conclude that the automatic prediction of salience shows very high performance. However this is perhaps not necessarily the case for all the frames of the video. And how to assess whether this algorithm provides performance above the others?

This is the second interest point in this tutorial: **how to measure the performance of an algorithm for automatic creation of saliency maps**.

You will calculate the automatic algorithm performance visual saliency extraction using three different metrics views on course.

1- PREDICTION OF SALIENCY MAP WITH DEEP CNN

In your working folder 'WorkSpace', download the zip file from the link and extract the archive:

http://dept-info.labri.fr/~schaabou/TD_saliency/section1/caffe.zip

Open a new terminal, and change the current directory to /opt/local/stow/caffe

Write this command:

```
> export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
```

```
> export LD_LIBRARY_PATH=/opt/local/stow/caffe/lib:$LD_LIBRARY_PATH
```

Execute the python code created to predict saliency map (add needed arguments).

```
> python ./python/ChaboNet4k_HOLLYWOOD_SaliencyGenerate.py
```

The prediction of saliency need times to be generated, to don't waste time let the program running and proceed to the second section of the TD.

2- RECOVERY OF THE DATA BASE AND THE CODE

Download the dataset and code to the following link : (here the predicted saliency map are already computed) and extract the archive in your working folder.

http://dept-info.labri.fr/~schaabou/TD_saliency/section2/tm_td4_saillance_sec2.zip

Description of the archive :

```
|— Dataset
|   |— Automatic_ChaboNet_saliency
|   |   |— actioncliptest_350
|   |   |— actioncliptest_451
|   |   |— Original_frames_half
|   |   |   |— actioncliptest_350
|   |   |   |— actioncliptest_451
|   |   |— Visual_Points
|   |   |   |— actioncliptest_350
|   |   |   |— actioncliptest_451
|— gbvs
|— Matlab_student
|   |— auc.m
|   |— calcAUCscore.m
|   |— calcCCscore.m
|   |— calcNSSscore.m
|   |— index_cliptest_350.txt
|   |— index_cliptest_451.txt
|   |— main.m
|   |— preview_saillance_et_fixations.m
```

The folder "Dataset" is composed of three files containing the frames of two videos "actioncliptest_350" and "actioncliptest_451" ("Original_frames_half"), the recorded eye fixations ("Visual_Points") and saliency maps automatically predicted ("Automatic_ChaboNet_saliency").

The folder containing the code and index files is called "Matlab_student".

The folder "gbvs" used to calculate new automatic saliency maps with the model of Itti.

3- FIRST TESTS

During this project you will gradually complete the main file "main.m". You can now open it and compile it (no visible effects for the moment).

The files used to calculate the metrics are already provided ("auc.m", "calcAUCscore.m", "calcCCscore.m", "calcNSSscore.m").

The "preview_saillance_et_fixations.m" file is provided to give you a graphical overview of the saliency maps and the points of interest.

4- READ THE INDEX FILE

The files "index_cliptest_350.txt" and "index_cliptest_451.txt" contain in each line the path to the automatic saliency, the visual points and the corresponding image. This is the link with the database that you need initially to open and read with Matlab. A good method is to open the file, store data in a variable (in the form of cells of "strings"), and close the file.

5- FIRST RESULTS ON AN IMAGE

At first, before launching the metric calculations on all the saliency maps, calculate AUC scores, PCC and NSS between the **first saliency image** and the corresponding of eye fixations.

To realize the value of this score, calculate the same scores between eye fixations and the saliency map having the same size and containing **random** integer values (between 0 and 255)

The theoretical values obtained for these metrics if we use a randomly model are:

- AUC : 0.5
- PCC : 0
- NSS : 0

Compare scores between these values and those that you get from the random model and then the saliency map generated by the algorithm.

6- MAIN LOOP AND RESULTS

Now create the main loop for calculating NSS scores, AUC and PCC between all the maps and the corresponding automatic eye fixations in the database.

Calculate and display the average values and the standard deviation associated with each of these metrics.

Use the function 'bar' to display the results in the form of three separate bars and 'errorbar' function to display error bars (standard deviation) on these bars. Do not forget to use 'hold on' and 'hold off'.

7- AND WITH RANDOM VALUES ?

Repeat the step 5 by substituting in this time each automatic saliency map by a map randomly generated on each time(as in step 4).

Calculate and display the average values and the standard deviation associated with each metric.

Compare with the values obtained in Part 5, and with the theoretical values.

8- COMPARISON WITH THE STATE OF THE ART

An automatic methods for creating saliency maps witch presents the more known method in the field is that of Laurent Itti. The folder "gbvs" contains ittikichmap function to compute the saliency of an image by this method.

Now open the file "calcul_itti_maps.m" and create the main loop to calculate and save the .jpg saliency Itti for all frames of the two videos.

Re-open the "main.m" file and complete in order to calculate the scores and figures for these new saliency maps. Compare scores with are calculated in parts 5 and 6.

9- REPORT

You are asked to write a concise report containing the following parts:

- Introduction
- Presentation of results and interpretation (how to evolve the performance and explain the reason).
- Conclusion (or problems encountered or resolved, possible prospects, ...)

Send your report, named MYNAME_Lab4.tar.bz2 , to souad.chaabouni@u-bordeaux.fr in an archive that also contains the Matlab code before midnight of Saturday, November 05, 2016.