# Boosting Saliency Prediction Accuracy with Temporal Information

*Master Semester Project Report*

Student: Ludo Hoffstetter
Supervisor: Bahar Aydemir

Image and Visual Representation Lab, EPFL

***Abstract*—Saliency prediction usually relies on training datasets exclusively consisting of saliency maps that are obtained by aggregating all fixation points into a single frame and convolving them. However, the unprocessed eye tracking data is clearly much richer and while aggregation is certainly an efficient way to filter out the noise, a more fine-grained approach can leverage the unused data for the benefit of accuracy. The idea of this project is to generate a new enhanced training dataset by retrieving the timestamps corresponding to each fixation point and dividing the observation span into equal time slices each forming a saliency heatmap.**

**We adapted the architecture of SimpleNet by adding a new loss for multi-duration saliency prediction so that the temporal data is injected in the network during training and is used for the final saliency map prediction. The results are quite conclusive as every accuracy metrics are being improved in comparison to the original SimpleNet model while keeping a simple architecture.**

## I. Introduction

The saliency of an image refers to the theoretic underlying spatial distribution of areas likely to capture the gaze of a human observer. It is typically represented as a heatmap of the same size as the referring image where each value is proportional to the corresponding pixel's likelihood of being observed.

Saliency prediction is a non trivial task as the model has to identify the features of an image most likely to attract the eye with appropriate weights. There is a variety of distinct approaches for this problem ranging from classic convolutional networks to object detection based techniques [1]. This area of research is essentially focused on the eye fixations that happen during a predetermined interval of time (typically 3-5 seconds) while discarding the order of appearance of the forementioned fixation points as well as their frequency through time. Ignoring temporal data is an efficient way to reduce the data flow and simplify the prediction task. However, there is another research field that is very similar and that does value the dimension of time: scanpath prediction [2]. A scanpath can simply be described as an array of fixation points sorted by order of appearance [3]. This methods requires sampling strategies in order to aggregate the eye tracking data of several observers for a given image. The research progress in this area is relatively slow compared to saliency prediction but it does provide insights that saliency interact with time [4].

The main idea of this project is to feed the extra temporal data used in scanpath prediction during the training of a saliency heatmap prediction model. The additional training data is constructed following the subsequent pipeline: first associate each fixation point produced by each observer with its timestamp, then group fixations in even time slices, aggregate each slice in a fixation map and finally 3D convolve all slice together with a gaussian kernel. The obtained result can easily be stored in $n$ heatmaps where $n$ is the number of time slices. By taking the same assumption as scanpath prediction, the data should be correlated with time as the first elements noticed by an observer are likely to be the most conspicuous while the gaze often focuses on details afterwards. This assumption will be analytically verified lately and in the mean time can be observed through an example (see figure 1).

Fig. 1: Saliency evolution through time (bottom) of an image (top). The observation span is divided into 5 equal time slices evolving from left to right.

In order to achieve a comparable result, we started from a recent saliency model that had competitive results as well as an open source architecture: SimpleNet [5]. This model's structure is neater, minimal and more interpretable than most solutions while still achieving state of the art accuracy on saliency benchmarks. The research team that developed this model claim that "SimpleNet is an optimized encoder-decoder architecture and brings notable performance gains on the SALICON dataset". The goal of this project is to increase SimpleNet's performance by leveraging the temporal information and adapting its architecture accordingly.

The rest of the report is organised as follows. In Section II, we describe the dataset used for the scope of this project. Section III contains the detailed explanation regarding how to recover the missing timestamps from the fixations. In section IV we analyze the temporal data. Section V presents how this new temporal information is incorporated to the SimpleNet model, and the results are discussed in Section VI. Finally, section VII concludes with a brief summary of our work.

## II. Data Description

Public saliency datasets are not numerous as eye tracking devices are not a widespread technology. Therefore the data is often collected by laboratories showing sequences of images to observers and recording their gaze. This method limits the number of produced samples since it does not scale well and cannot make use of crowdsourcing.

However, the SALICON dataset proposed an alternate scalable method for saliency data collection that consists of emulating the behavior of human eyes by blurring the observed image outside of a circle centered on the mouse cursor [6]. That way the subject has to move the mouse to clearly be able to see the image, and those movements are recorded and interpreted as an estimated gaze trajectory. This technique is compatible with crowdsourcing as it does not require any external device except a mouse or a trackpad.

The final dataset is a mix between lab experiments and crowdsourcing, both using the mouse approximation. The captured mouse samples are normalized to 100 Hz to homogenize the data. The pre-processed samples are aggregated by image and the fixation annotations are then extracted with ClusterFix: a K-means clustering using distance, velocity, acceleration and angular velocity. Finally, the saliency maps are obtained by simply blurring the fixation map with a Gaussian filter and normalizing.

SALICON is the most substantial saliency

dataset thanks to this scalable data collection method. It offers 10000 training samples as well as 5000 validation samples. The images are all sampled from the 2014 COCO dataset and share the same dimension: 480 by 640 pixels. Besides the saliency heatmaps, SALICON also provides the recorded raw gaze points for each observer where each point has an associated timestamp, as well as the list of fixation points sorted by time but with no timestamp.

We are using SALICON for the entirety of this project because of its considerable amount of samples and also because of the extra data it offers.

## III. Fixation Timestamp Retrieval

As stated earlier, fixation points do not have a timestamp value and there is no one to one matching with raw gaze points as fixation points are obtained with K-means and not K-medoids. The challenge here is to find a way to accurately recover the missing temporal information from fixations.

A first intuition for this task is to simply search the closest gaze point to the current fixation and select its timestamp value. This method generally works well but it is limited by the fact that observers often explore the image back and forth, and therefore two close gaze point could have significantly different timestamps. The explicit formula is the following:

$$f_{ts} = \underset{p_{ts},\, \forall p \in GazePoints}{\operatorname{argmin}} ||f_{xy} - p_{xy}||^2$$

The previous approach does not take into account the fact that the list of fixations of a given observer are sorted by order of appearance. Knowing that, one can approximate the timestamps by simply assuming that all fixation points are evenly distributed in time. Knowing that the observation recording timespan is equal to 5000 ms in SALICON, we have the following formula:

$$f_{ts} = 5000 * (index + 1)/(list\ size + 1)$$

Both techniques are incomplete and do not consider all of the available clues. However, we could obtain a robust timestamp estimator by combining them together. This is done by defining a score function for each gaze point relative to the current fixation:

$$p_{score} = ||f_{xy} - p_{xy}||^2 + w * |p_{ts} - f_{ets}|$$

*Where $f_{ets}$ is the estimated timestamp computed with the second method and $w$ is the balancing coefficient between the two techniques.*

Each fixation timestamp is now estimated by selecting the timestamp of the gaze point with the lowest score. But we still have to fine tune the balancing coefficient to optimize the result. In order to define the quality of a timestamp prediction we define two measures to help us. The first one is simply the pixel distance between the fixation point and the selected gaze point. Obviously this measure should be as low as possible as the fixation timestamp is correlated with near gaze points. The second metric measures the quality of the prediction by using the initial sorted list of fixations. If we sort the fixations according to their predicted timestamp, we want the order to be as close as possible with the original one. This is done by computing the mean down edges per observer, as explained with an example in figure 2). The maximum value is 0, meaning that the order did not change. A negative value of large magnitude indicates a poor prediction.

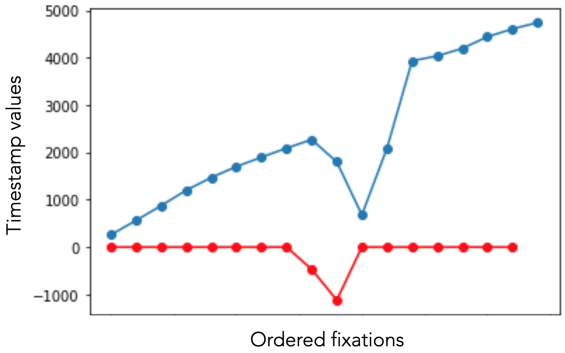


Fig. 2: Predicted timestamp values of a given observer sorted in the original fixation order (in blue). The convolution of a filter isolates the down edge (in red) and the value of every point is summed.

Now that we have defined two measures, let's compute them for a range of $w$ values (cf figure 3). The optimal values seems to be for $w = 0.006$ where both metrics are reasonably low and form a local minimum. With the previous formula and the

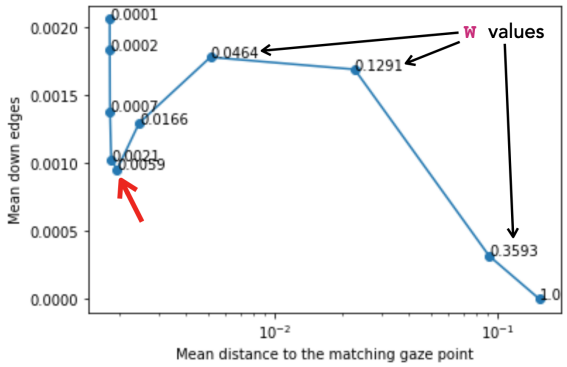


Fig. 3: Mean pixel distance per mean down edges for $w$ values where both measures are averages over the entire training dataset

optimal value of $w$ we can now accurately recover the temporal data for each fixation.

## IV. Temporal Data Analysis

Let's take a look a the freshly recovered temporal information by first verifying that saliency is indeed correlated with time. Figure 4 presents the timestamp frequencies with a histogram. Observers clearly seem to produce more fixations around one second which would correspond to the initial search for context in the image while subsequent timestamp frequencies are linearly decreasing probably because the eyes are exploring less and are more contemplative.

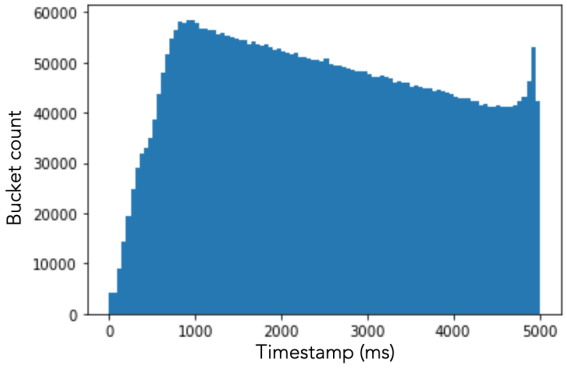


Fig. 4: Histogram of timestamp frequencies

This result is confirmed by figure 5 that displays the average saliency maps of the observation timespan divided into 5 equal time slices, i.e. one per second. Indeed, during the first second fixations are widespread and slightly skewed to the left, indicating that observers tend to look at a wide portion of the image. The next second, the saliency map is quite similar but with a tendency to the right this time (this artefact may be due to the fact that the data was collected in western countries, where people read from left to right). Finally, the fixations migrate to the middle of the image, this is probably correlated with the fact that pictures are generally centered on the subject.

Overall this analysis proves indeed that saliency changes through time in a meaningful way, just as we suspected initially.

## V. Leveraging Temporal Data

Let us define first the original SimpleNet architecture before introducing the modifications. From its original paper [5], SimpleNet is described as a single-stream encoder-decoder architecture that predicts the pixel-wise saliency values. The network is fully convolutional and the architecture is shown in its entirety in figure 6.

The readout architecture is composed of two convolutional layers, the first being combined with ReLU, and the second with a sigmoid function to finally output the saliency heatmap. During training, SimpleNet uses as a loss a combination of Kullback-Leibler Divergence (KLdiv) and Pearson Cross Correlation (CC).

The issue with the additional generated temporal information is that it is available only for training samples, meaning that it cannot simply be fed to the model as input features. Our solution was to add a second loss comparing the predicted time sliced saliency with the temporal data. Our model therefore produces two outputs: $n$ successive saliency time slices where $n$ can be any value selected during training, and the classic saliency heatmap. The network is described in figure 7. The new architecture adds two new readout layers: one at the same level as the original one that generates the predicted saliency slices, and one that mixes the original SimpleNet prediction with the $n$ saliency slices. By doing so the model fully utilizes the potential of SimpleNet and propagates the temporal information gain in the network with the second loss. This loss is computed by averaging the $n$ CC and KLdiv measures of each matching pair of prediction and ground truth slices.

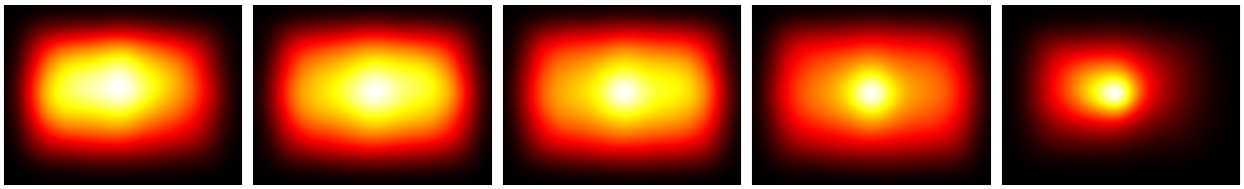

Fig. 5: Evolution of the normalized average saliency map through time

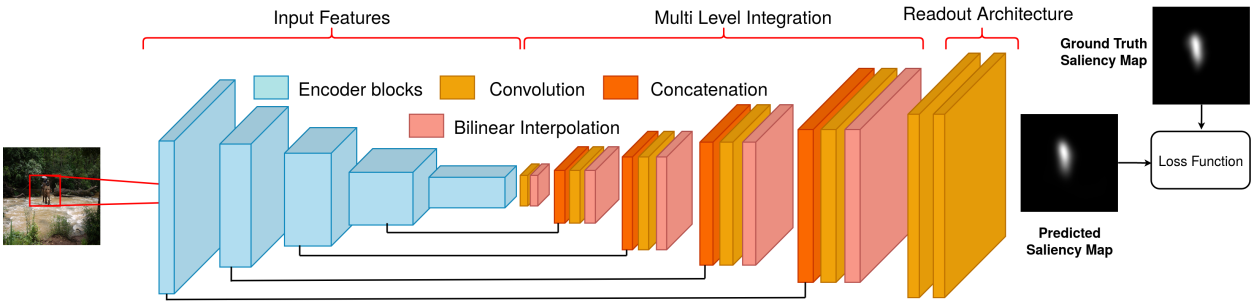


Fig. 6: SimpleNet's architecture

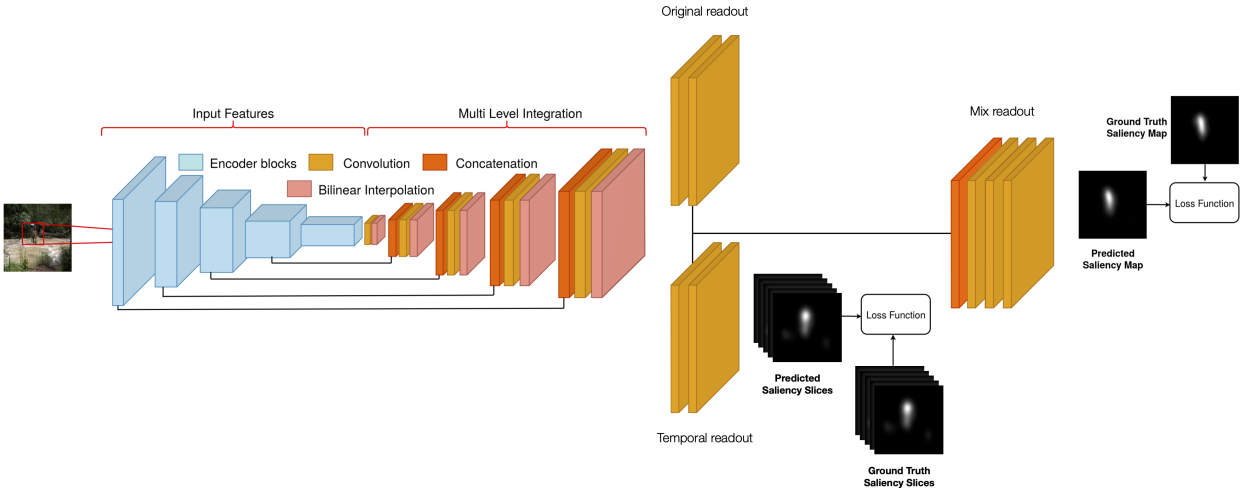


Fig. 7: Our network's architecture

The benefit of using a second loss is that the model can thus also be used to predict the temporal saliency maps of any input image.

## VI. Evaluation

For this section we trained the model with several configurations for the loss combination, the temporal loss coefficient and the number of time slices. The results were then obtained by submitting the predictions of SALICON's test set to the corresponding CodaLab competition. Each configuration's associated accuracy metrics are displayed in table I [7].

The results clearly prove the efficiency of temporally boosted saliency models since every accuracy measures of SimpleNet are outperformed. Specifically, the model trained with 5 time slices holds the local record of almost all metrics and significantly increases SimpleNet's accuracy.

CodaLab's ranking is based on the SAUC metric. The model configuration achieving the local maximum SAUC score attained the 4th place while SimpleNet stands at the 14th position. Further ranking

| Model | Temporal loss coefficient | Time slices | AUC | CC | KLdiv | SAUC | IG | NSS | SIM |
|---|---|---|---|---|---|---|---|---|---|
| SimpleNet | - | - | **0.869** | 0.907 | 0.201 | 0.743 | 0.880 | 1.960 | 0.793 |
| KL+CC | 0.05 | 10 | 0.867 | 0.906 | 0.212 | 0.743 | 0.884 | 1.972 | 0.792 |
| KL+CC | 0.1 | 5 | **0.869** | **0.909** | **0.199** | 0.741 | **0.890** | 1.973 | **0.796** |
| KL+CC | 0.1 | 10 | 0.868 | 0.906 | 0.207 | 0.745 | 0.885 | **1.978** | 0.793 |
| KL+CC | 0.5 | 10 | 0.868 | 0.906 | 0.203 | 0.744 | 0.870 | 1.949 | 0.792 |
| KL+CC+NSS | 0.05 | 10 | 0.868 | 0.905 | 0.206 | **0.746** | 0.885 | 1.955 | 0.793 |
| KL+CC+NSS | 0.25 | 10 | 0.868 | 0.906 | 0.204 | 0.744 | 0.883 | 1.974 | 0.792 |

TABLE I: Accuracy results for several model configurations

| Model | Temporal loss coefficient | Time slices | AUC | CC | KLdiv | SAUC | IG | NSS | SIM |
|---|---|---|---|---|---|---|---|---|---|
| SimpleNet | - | - | **1** | 3 | 3 | 6 | 4 | 12 | 6 |
| KL+CC | 0.1 | 5 | **1** | **2** | **2** | 8 | **2** | **10** | **4** |
| KL+CC+NSS | 0.05 | 10 | 2 | 4 | 4 | **3** | 3 | 14 | 6 |

TABLE II: CodaLab rankings of the best configurations

comparison are stated in table II.

Each parameter could be fine-tuned in order to achieve an even more impressive accuracy gain, but it would require some time and good computational resources.

## VII. Conclusion

Saliency prediction is not an obvious task and choosing the right model demands a lot of experimentation. By exploring the recent related papers and looking for the most promising methods, we ended up having a very accurate predictor by using an adapted version SimpleNet that is exploiting the recovered temporal data. This project proves the importance and the efficiency of temporal data and leaves the door open to further usages in the sake of prediction accuracy.

## References


[1] Xavier Giró-i-Nieto Marta Coll Pol and Kevin Mc Guinness. The importance of time in visual attention models. August 2017.

[2] Kevin McGuinness Marc Assens, Xavier Giro-i-Nieto and Noel E. O'Connor. Saltinet: Scan-path prediction on 360 degree images using saliency volumes. August 2017.

[3] Kevin McGuinness Marc Assens, Xavier Giro-i-Nieto and Noel E. O'Connor. Pathgan: Visual scanpath prediction with generative adversarial networks. Computer Vision Fundation, 2018.

[4] Alan Kingstone Nicola C. Anderson, Fraser Anderson and Walter F. Bischof. A comparison of scanpath comparison methods. Behavior Research Methods, December 2014.

[5] Pradeep Yarlagadda Navyasri Reddy, Samyak Jain and Vineet Gandhi. Tidying deep saliency prediction architectures. March 2020.

[6] Juanyong Duan Ming Jiang, Shengsheng Huang and Qi Zhao. Salicon: Saliency in context.

[7] Aude Oliva Antonio Torralba Zoya Bylinskii, Tilke Judd and Fredo Durand. What do different evaluation metrics tell us about saliency models? March 2019.