The Importance of Time in Visual Attention Models

Degree's Thesis
Audiovisual Systems Engineering

Author: Marta Coll Pol
Advisors: Xavier Giró-i-Nieto and Kevin Mc Guinness

Dublin City University (DCU)
2017 - 2018

Abstract

Predicting visual attention is a very active field in the computer vision community. Visual attention is a mechanism of the visual system that can select relevant areas within a scene. Models for saliency prediction are intended to automatically predict which regions are likely to be attended by a human observer. Traditionally, ground truth saliency maps are built using only the spatial position of the fixation points, being these fixation points the locations where an observer fixates the gaze when viewing a scene. In this work we explore encoding the temporal information as well, and assess it in the application of prediction saliency maps with deep neural networks. It has been observed that the later fixations in a scanpath are usually selected randomly during visualization, specially in those images with few regions of interest. Therefore, computer vision models have difficulties learning to predict them. In this work, we explore a temporal weighting over the saliency maps to better cope with this random behaviour. The newly proposed saliency representation assigns different weights depending on the position in the sequence of gaze fixations, giving more importance to early timesteps than later ones. We used this maps to train MLNet, a state of the art for predicting saliency maps. MLNet predictions were evaluated and compared to the results obtained when the model has been trained using traditional saliency maps.Finally, we show how the temporally weighted saliency maps brought some improvement when used to weight the visual features in an image retrieval task.
The code used during the development of this project can be found at `https://github.com/imatge-upc/saliency-2018-timeweight`.

Acknowledgements

I want to thank my advisors Xavier Giro-i-Nieto and Kevin Mc Guinness for all the help and guidance during the development of this project.

I would also like to give a special thanks to Marc Assens and Alejandro Woodward for helping me through my learning process in the programming languages used in this project. Eva Moehdano for her contribution to our project, and Andrea Calafell to let us use her project's template.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Visual attention models

Computer Vision is the field that studies how computers can gain high-level understanding from digital images or videos. Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images. The methods for vision understanding are usually inspired by how the human visual system works [15]. This is the case in the study of semantic segmentation, object recognition or saliency prediction (also called visual attention) among others [3].

Thanks to the increase in the amount of available computation and data in recent years, computer vision research has moved to the use of deep learning to solve many complex tasks. Deep learning allows computational models to learn representations of data with multiple levels of abstraction. These computational models are layer-based and have several nodes per layer, also called neurons, which compute basic operations. During the learning process, the back-propagation algorithm is used to indicate how the parameters involved in these basic operations should change and this process is repeated until the model has properly learned how to perform the desired task. These methods have significantly improved the state-of-the-art in many tasks like speech recognition, visual object recognition, object detection or saliency prediction (visual attention)[26]. This project is focused on the task of saliency prediction and deep learning tools are used.

Visual attention is a mechanism of the visual system that allows humans to selectively process visual information of certain areas, which are considered of interest within the visual field, while ignoring other perceivable information[32]. This mechanism helps humans gain an understanding of what's going on in their visual field. These areas or locations where a person fixates the gaze for a while, are those that usually contain objects that can be interacted with, text, people, or are areas where actions are happening [10], etc.

Visual attention models aim to predict which areas of an image/frame are most likely to be selected in the viewing process by a human observer. Humans have a limited capacity for processing information. At any given time, only a small amount of information available on the retina can be processed and used in the control of behavior [13]. If we can teach computers where to look, the amount of data to process can be reduced in the same way as in the human visual system. In other words, visual attention models can be used to filter the relevant information and save computing resources in addition to improve other computer vision tasks such as object recognition [3] [25].

1.1.1 Saliency information representations

There are different ways of collecting eye-movements data while a series of images are being shown. The most reliable way is to use eye-trackers[42], but since there is an increasing need

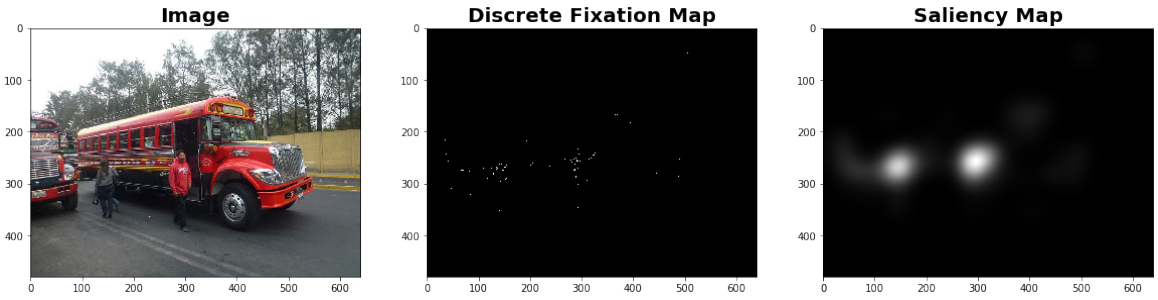

Figure 1.1: Saliency information representations of an example image

of large datasets with saliency annotations, a cheaper method based on mouse-trackers has also been proposed [17]. This method consists of asking observers to select the areas from an image that catch their attention.

From the eye- or mouse-data collected, fixation points are extracted. Fixation points are those locations where human observers fixate their gaze for a while. Merging fixation points of all observers for a given image creates what is called a discrete *fixation map* (see Figure1.1); this kind of maps are used in location-based metrics for the evaluation of saliency models' results[9]. Saliency maps represent the areas that attract more visual attention under the form of a heatmap with the same dimensions as the image under analysis. *Saliency maps* are two-dimensional arrays where a location with a higher scalar value means that it is more likely to attract human attention (see Figure1.1). Saliency maps are typically built by convolving a discrete fixation maps with a Gaussian kernel [28].

1.2 Motivation

The research in the saliency prediction field has been influenced in recent years owing to the resurgence of neural networks in the computer vision community. Consequently, models have significantly driven up performance scores. At first sight, predicted saliency maps from top models might look close to ground truth maps. Nevertheless, recent work [10] reveals that state of the art models still present several limitations. Figure 1.2 shows some examples from [10] where models do not perform as expected.

Accepting that state of the art models still offer room for improvement, our hypothesis is that encoding temporal information in the ground truth maps could help improving results for the case of images where temporal information seems to matter. To better depict our assumption, Figure 1.3 includes a particular image in the sense that it has only few regions of interests. We can tell that the model is able to predict the most relevant area of the image, but misses the salient regions on the top right corner.
According to the paper[37], "In any scene, it is likely that there will be only a few locations of extremely high salience. These will be selected first during viewing and the limited number of such locations means that there will be a high degree of consistency in the locations selected early in viewing by all observers. Conversely, there are likely to be quite a number of locations with similar, moderately salient characteristics. Hence once the high salience locations have been visited, there exists a much broader range of possible saccade targets. If the oculomotor system selects from among these possible targets at random, then this would give rise to a lower degree of consistency between observers as viewing progress". Figure 1.4 depicts this effect. If it is

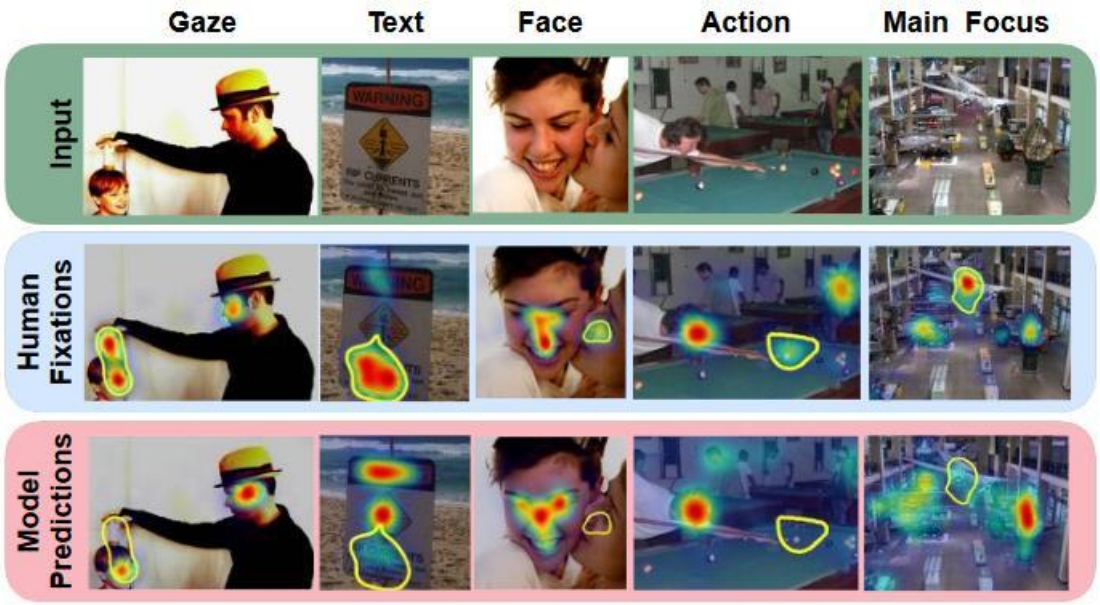

Figure 1.2: Detected cases where models can still make significant improvements. High- density regions of human fixations are marked in yellow and show that models continue to miss these semantically-meaningful elements[10].

proved that later fixations tend to be selected randomly and will vary depending on the observer, we cannot expect a model to learn to predict this randomness.
Specially in those images with few relevant locations models seem to be able to predict early fixations but have difficulties predicting later ones. For this kind of images, the observation time given to the observers, during ground truth data collection, might be too long for the few relevant things to see during visualization, causing many fixation points collected to have this random behavior. For this reasons, we consider that using a ground truth that considers equally all fixation points could be adding an unnecessary noise to the model's input, obstructing its learning process.

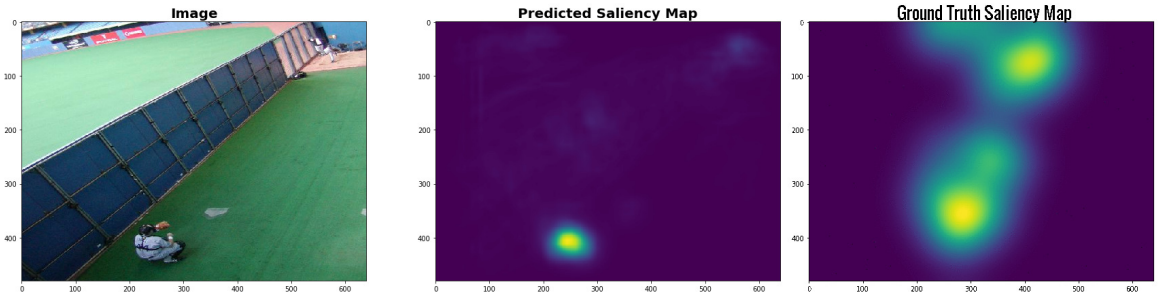

Figure 1.3: On the right, there is the ground truth saliency map for the example image, which is provided in the dataset. In the center the predicted saliency map from a state-of-the-art model called MLNet[12] that performed well in the MIT saliency benchmark[8]. In the example, we can observe an image where early fixation points are correctly predicted by MLNet, a model for saliency prediction, while later fixation points have significantly less saliency in the predicted map compared to the ground truth Saliency Map.

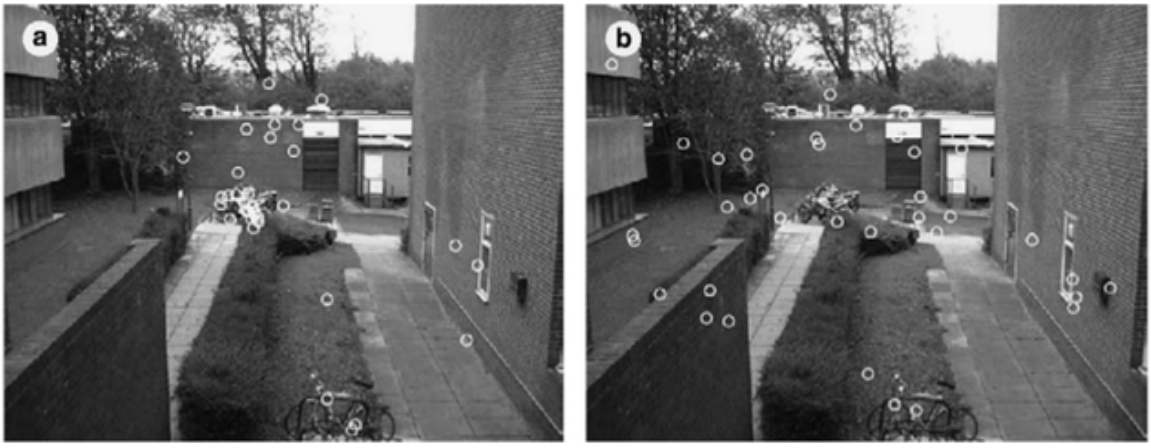

Figure 1.4: Fixation locations (indicated by circles) for all observers combined (a) during the first second after stimulus onset, and (b) during the fifth second after stimulus onset, for one of the images viewed. There appears a greater degree of consistency in the locations chosen early in viewing than several seconds later[37].

1.3 Statement of purpose

This project proposes *Temporally Weighted Saliency Maps (TWSMs)*, a novel type of ground truth saliency map that takes into account the increasing randomness in time of the eye fixations. This saliency representation is inspired by the one presented in [5], where a weighting function is used to give more weight to early fixation points than later ones.

Our work has been structured with the following milestones:

- Obtain sorted fixation points from two scientifically accepted datasets: SALICON (mouse-based) and iSUN (eye tracker-based).
- Propose an approach to temporally weight the fixation points and build Temporally Weighted Saliency Maps (TWSMs).
- Evaluate a pre-trained state of the art model (MLNet [12]) over unweighted Normal Saliency Maps (NSMs) and Temporally Weighted Saliency Maps (TWSMs) to assess the impact of the proposed modification.
- Re-train MLNet over NSMs to replicate the results published in [12].
- Train MLNet using WSMs as ground truth.
- Evaluate the two versions of MLNET over the two types of saliency maps (NSMs and WSMs).
- Compare NSMs and TWSMs to solve a third computer vision task aimed at visual object retrieval.

1.4 Technical skills acquired during this work

This research project has represented the first contact of the main author with deep learning. As many people from the research community would advice, the easiest way to introduce yourself to neural network architectures is through the use of a high-level neural networks API like Keras. This API allows to speed up experimentation. Keras is written in Python and is capable of running using TensorFlow, CNTK or Theano as a back-end. For this reasons, and for the amount of documentation available, we decided that the best choice would be the use of Python as the programming language to code and the Keras API to speed up experimentation. The saliency prediction model called MLNET [12], which has been used as a reference model for visual saliency prediction, was chosen following this criterion. This model is implemented in Python, uses the Keras API and Theano back-end.

1.5 Work Plan

While most of this research was developed at Dublin City University (DCU) during the Spring semester of 2018, the project actually started earlier during Autumn 2017 at the Universitat Politecnica de Catalunya (UPC). During that semester, I started attending to the research meetings on visual saliency research of the team formed by Marc Assens, Dr. Xavier Giro-i-Nieto (from UPC) and Dr. Kevin McGuinness (from DCU), mainly on videocalls. As a result I became familiar with the saliency prediction field and had an introduction to the two time-aware representations, the *Weighted Saliency Maps* and *Saliency Volumes*, they had published in [5].

This core of the project was developed at DCU during Spring 2018 with the following work plan, with a few exceptions and modifications explained in the section 1.6.

1.5.1 Work Packages

Task name	Start date	Due date	Duration (days)
Part 1	01/02/18	07/03/18	34
Research for addressing the first task	01/02/18	09/02/18	8
Choice of the dataset and model for Saliency Prediction	05/02/18	06/02/18	1
Prepare dataset suitable for the chosen model	07/02/18	09/02/18	2
Obtain ordered fixations	11/02/18	13/02/18	2
Obtain model predictions for the chosen dataset	14/02/18	16/02/18	2
Dataset Study	18/02/18	19/02/18	1
Create weighted saliency maps, and normal SM ground truth	18/02/18	23/02/18	5
Process data to be suitable for evaluation	25/02/18	28/02/18	3
Evaluate first task results	04/03/18	06/03/18	2
Consider other ways to generate wSM	07/03/18	11/03/18	4
Create a image set for testing purposes	06/03/18	07/03/18	1
Part 2	12/03/18	03/04/18	22
Research about used architectures	12/03/18	14/03/18	2
Define strategy for improving a model	15/03/18	18/03/18	3
Choice of a model to fine-tune	18/03/18	19/03/18	1
Prepare dataset suitable for the chosen model	20/03/18	23/03/18	3
Train the model with the testing set obtained in Part 1	26/03/18	29/03/18	3
Evaluate results	29/03/18	30/03/18	1
Evaluate performance improvements	01/04/18	03/04/18	2
Part 3	04/04/18	22/04/18	18
Research on the existing metrics	04/04/18	08/04/18	4
Define a strategy to create the new metric	08/04/18	11/04/18	3
Write the code for the new metric	12/04/18	18/04/18	6
Test the created metric	18/04/18	22/04/18	4
Part 4	22/04/18	11/05/18	19
Evaluate Part 2's results using the new metric	22/04/18	24/04/18	2
Perform any necessary improvement	24/04/18	09/05/18	15
Conclusions study	09/05/18	11/05/18	2
Writing of the tesis	01/03/18	24/05/18	84

Figure 1.5: Worck Packages presented in the first Report

1.5.2 Gantt Diagram

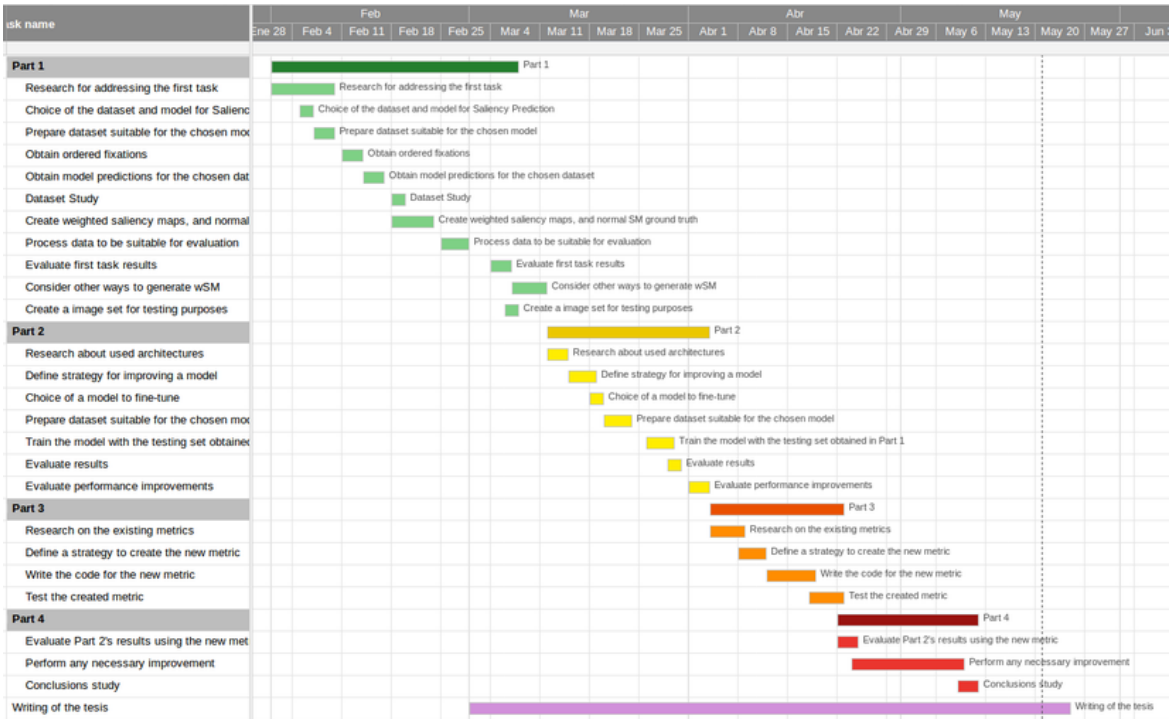

Figure 1.6: Gantt Diagram of the Degree Thesis presented in the first report

1.6 Incidents and Modification

Even though we proved all the hypothesis being tested, due to complications when trying to replicate MLNet's published results, we ended up without enough time to perform the last part of the project where we wanted to create a new metric for the evaluation of our weighted maps. Instead an evaluation of Temporally Weighted Saliency Maps was performed for the visual search task.

Chapter 2

Literature Review

2.1 Deep Learning

Neural Networks are hierarchical structures made from basic structures called neurons. Neurons receive several inputs, each input is weighted and a simple operation is performed in the neuron followed with an activation function to compute an output (See Figure 2.1). A layer is formed when several neurons are clustered together. A neural network is composed of the input layer, at least one hidden layer and the output layer, as exemplified in Figure 2.2. The layers present between the input and the output layers are called hidden layers and each hidden layer is usually the input of the following layer[2].

When training a network to solve a specific task, in the beginning, weights are initialized, either randomly if the training is performed from scratch or with already trained weights for fine-tuning. The step that follows is checking the model's performance with the initial weights. The input is passed through the network and an output is calculated. This step is called forward-propagation since the flow goes from the input to the output of the network. At this stage, what we have is the actual output of the network and the desired output. The metric used to evaluate performance is called loss function, and it measures how well the neural network is able to reproduce the desired output. To simplify things, we can define the network's goal as the minimizing of the loss function. To minimize this function weights are optimized using the Back Propagation algorithm[27] which starting from the output layer and moving towards the input, updates layers' weights. After updating the weights, performance is tested again. The whole process is repeated over and over again until convergence is achieved [4].

As an introduction to Deep Learning models, we are going to present Convolutional Neural Networks (CNNs), architectures that have allowed significant advances on the state-of-the-art in computer vision tasks such as object detection [14] or image classification[23].

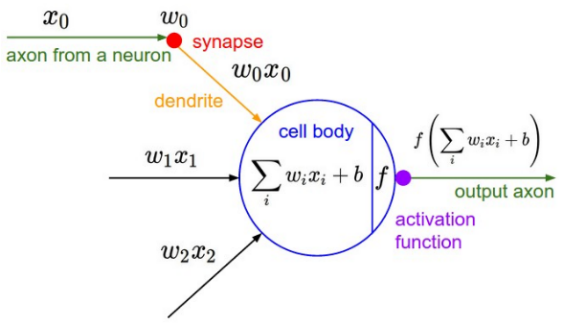

Figure 2.1: Basic structure called neuron that given a certain amount of inputs, performs a basic operation to compute an output[2].

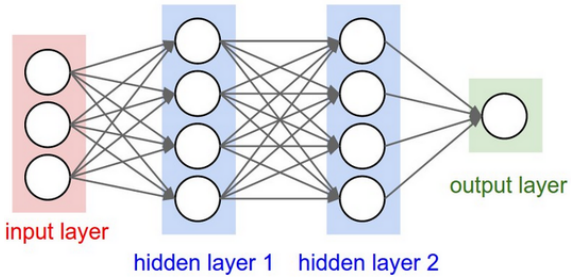

Figure 2.2: This Figure shows a three-layer neural network (two hidden layers of four neurons each and a single output), with three inputs[2].

2.1.1 Convolutional Neural Networks

Convolutional Neural Network (CNN) architectures receive a multi-channel image as an input and are usually structured with a set of Convolutional layers each followed by a non-linear operation (usually RELU) and sometimes by a pooling layer (usually max-pooling). At the end of the network, Fully Connected layers are usually used [23][6].

Each Convolutional layer comprises a set of independent filters, let's say for instance that in a layer we have 6 filters, each filter is going to be convolved with the input image to produce what is called a feature map, so in this example, we would obtain 6 feature maps. These filters are initialized randomly and become the parameters that the model is going to learn during the training process using the Back Propagation algorithm [27], to automatically adjust to the task being solved.
What is particular of Convolutional layers is that for a particular feature map, each neuron is connected only to a small chunk of the input image, forming matrices. The use of this layer is really efficient, especially for computer vision tasks since the amount of parameters required to be trained is significantly smaller compared to the ones required for the Fully Connected layers.

Pooling layers' function is to reduce the number of parameters by progressively reducing the spatial size of the representations of the image. Therefore, reducing the computation needed for the following layers.

Fully Connected layers are layers where each neuron receives all the outputs of the previous layer. All these connections mean a lot of parameters to be trained. Therefore more computation resources are required for these layers compared to the ones needed for the Convolutional ones. In the case of image classification tasks, Fully Connected layers are the ones whose parameters are trained towards differentiating each image class from the feature maps.

2.2 Saliency Prediction

The first predictive works were biologically inspired, Itti *et al.*[16], implemented an architecture for visual saliency prediction in images, that extracted low-level visual features based on colour, intensity and orientation which were inspired in Koch *et al.*[22] feature maps. These features were then integrated to form a saliency map, an image in which the intensity of each pixel indicates the probability of the corresponding pixel in the original image to be fixated by a human

observer. Experiments showed difficulties in complex scenes due to the simplicity of feature maps. In [20, 41], they also proposed to predict fixations using local low-level features.
Torralba *et al.*[38] showed how high-level semantics or global contextual information, also attracts humans attention and those can be used to improve predictions of observers' eye fixations. Later, Judd *et al.* [19] proposed a model that combined low-level features (color, orientation,..) and high-level features (such as objects) to predict fixations.

The advances in deep learning lead to the first attempt to train a model for Saliency Prediction using Deep Convolutional Networks, which was made by Vig *et al.* [40]. But it was the introduction of AlexNet[23] in the ImageNet challenge[34] for large-scale visual recognition, that allowed models trained for different computer vision tasks to jump to the current state-of-the-art. The main problem of visual attention models was the limited amount of training data, to help deal with this fact, Kümmerer *et al.* [24] presented the model DeepGaze, which introduced a novel way of reusing models trained for other computer vision tasks to be applied in saliency prediction. DeepGaze was built on top of AlexNet and results outperformed all state-of-the-art models at the time it was published in the famous MIT saliency benchmark[8]. In addition, it gave new insights in the psychophysics of fixation selection.
The appearance of large datasets for saliency prediction like SALICON[17] which collected data using a mouse-tracking system, lead to the emergence of a number of other neural networks models. An example would be the model chosen for our experiments, MLNet[12], which was built on top of the VGG network[36] and trained on the SALICON dataset and later fine-tuned on MIT300 dataset. Pan *et al.*[31] also trained two different architectures on the SALICON dataset, a Shallow Deep Convolutional Network which was trained from scratch, and a deeper one whose first three layers were adapted from VGG[36], which was trained for image classification.

Recent advances in deep learning such as the Generative Adversarial Networks (GANs) have also been applied to saliency prediction. GANs architecture consists of two modules, a generator and a discriminator. While the generator learns how to predict data with the same structure as the data it is shown at the input, the discriminator learns to tell the difference between generated data and real data. Once the discriminator has learned to tell a difference, it can be used by the generator to improve its predictions. Pan *et al.*[30] introduced SalGAN, a Deep Convolutional Neural Network that was trained using adversarial examples. The first stage of the network consists of a generator that learns from downsampled versions of saliency maps. Generator results are then processed by a discriminator trained as a binary classifier to discriminate between generated maps and ground-truth maps. Results proved that the state-of-the-art can be achieved for different metrics when a model is trained using artificial saliency maps.

2.2.1 MLNet

To perform our experiments we have used a saliency prediction model, called *MLNet* [12], that scored well at the MIT saliency benchmark[8]. While many state-of-the-art models for saliency prediction employ fully Convolutional networks that perform a non-linear combination of features extracted from the last layer to predict saliency maps. MLNet proposed a different architecture that combines a CNN with 13 fully Convolutional layers to compute low, medium and high level feature maps from the input image, that are extracted from different layers, followed by an Encoding network which taking feature maps extracted as its inputs, learns a feature weighting function to generate saliency-specific feature maps and produces a temporary saliency map. Afterwards, a prior learning network (See Figure 2.3) is applied to produce the final saliency map prediction[12]. Since MLNet architecture is built on top of the VGG model[36], weights were

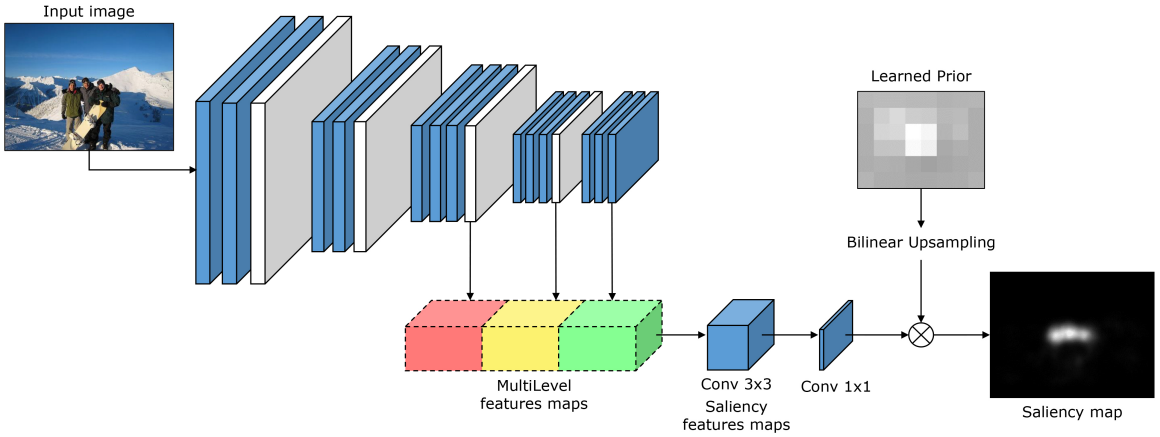

Figure 2.3: Overview of MLNet. A CNN is used to compute low and high level features from the input image. Extracted features maps are then fed to an Encoding network, which learns a feature weighting function to generate saliency-specific feature maps. A prior image is also learned and applied to the predicted saliency map [12].

initialized with the pre-trained ones from VGG-16. Then the model was trained first using the 2015 version of the SALICON dataset to evaluate the model's performance in the SALICON 2015 challenge and later it was fine-tuned using MIT300 dataset to submit results to the MIT saliency benchmark. Our work has been based on the version that was trained using only the SALICON dataset.

Chapter 3

Datasets of temporally sorted fixations

3.1 Introduction

In this section, we introduce the two well-known datasets for saliency prediction, *iSUN* and *SALICON*, that have been used in different stages of our work. We are also going to explore the methods used to obtain fixation points in order of visualization since knowing fixations order was necessary to be able to perform our experiments.

3.1.1 iSUN

The iSUN dataset[42] was build using the images from the SUN database. For each image, they provide the image content in JPG, image resolution, scene category, and saliency ground-truth data composed of gaze trajectory with timestamps for each location, and fixation points. Ground-truth data was collected through Amazon Mechanical Turk[39] by using gaze-tracking in web-cam videos recorded from participants observing the images they were shown. The dataset is divided into the three classic partitions; training, validation and test sets. Our experiments were performed using only the training set which contains 6,000 images.
Fixation points are obtained applying a clustering algorithm called *Mean-shift*[1] using gaze trajectory locations as original data points.

3.1.2 SALICON

Whereas iSUN collects ground-truth data using a gaze-tracking system to record viewing behaviours, SALICON used a mouse-tracking system instead. They designed a new mouse-contingent multi-resolutional paradigm which is based on neurophysiological and psychophysical studies of peripheral vision to stimulate the natural viewing behaviour of humans[17]. Comparisons on the OSIE dataset showed that the two tracking systems generated highly similar saliency maps at the output. To enable large-scale data collection the experiment was deployed on the Amazon Mechanical Turk[39].

For each image, they provide the image content in JPG, image resolution and ground-truth data which includes mouse trajectory with the corresponding timestamps for each location, and fixation points. The dataset is divided into the training set which contains 10,000 images, and the validation and test sets, each one containing 5,000 images. Since ground-truth data for the test set is not publicly available we performed the evaluation of our results using the validation set.

[1] More details in Section 3.1.2.1

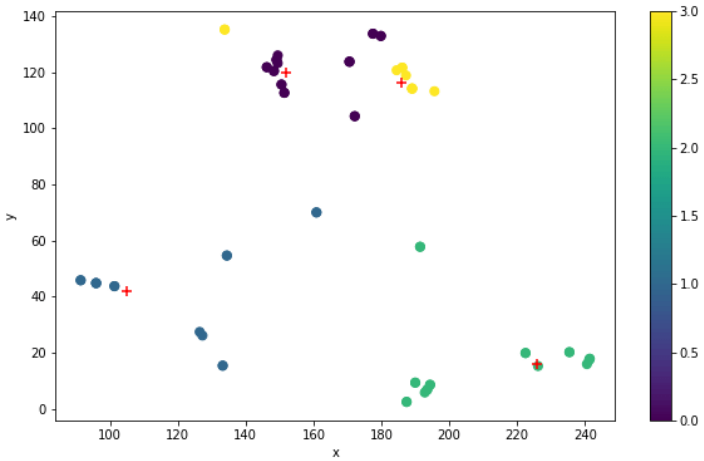

Figure 3.1: Example of Mean-shift result applied in a selected image from the iSUN dataset for a given observer. Data points with the same color belong to the same cluster and the centroid for each cluster is represented with a red cross.

3.1.2.1 Sorted iSUN

As we have previously seen in section 3.1.1, iSUN creators used gaze-tracking systems to collect gaze trajectory locations and applied a clustering algorithm called Mean-shift to obtain fixation points from those locations. The use of this method gives no warranties that fixations points are listed ordered in the ground-truth data provided in the dataset. Therefore, we had to replicate the same method that the authors used to obtain them in order.
Mean-shift is a clustering algorithm that assigns a group of data points to clusters iteratively shifting them towards the mode. In statistics, the mode of a set of values is the value that appears more often, in this case, it can be understood as the highest density of data points. The center of each cluster is called centroid and it is the arithmetic mean position of all points within the cluster[11](see Figure 3.1). For iSUN, fixation points are the resulting centroids after applying Mean-shift using gaze locations as the algorithm original data points.
When replicating the method used, we associated each gaze location with its corresponding timestamp. After applying the algorithm, we used the timestamps mean of all data points within each cluster as a weight to retrieve fixation points in order[2].

3.1.2.2 Sorted SALICON

While iSUN uses an eye-tracking system, the way of extracting fixation points from gaze locations is completely different for SALICON. As previously mentioned in section 3.1.2, SALICON used a mouse-tracking system to capture the gaze trajectory. This way of obtaining the data allows them to simply exclude half of the samples with high mouse-moving velocity for each observer while keeping the fixation points[17]. While for iSUN we had reasons to believe fixation points are not given in order of visualization in the dataset, it is not the case for SALICON. To test if they were already given in order of visualization, we decided to plot fixation points labeled with a number corresponding to the order of which they are given in the dataset, for some images and their corresponding saliency maps.

[2]We used the Mean-shift implementation found in `https://github.com/mattnedrich/MeanShift_py` using a multivariate Gaussian kernel to replicate the method

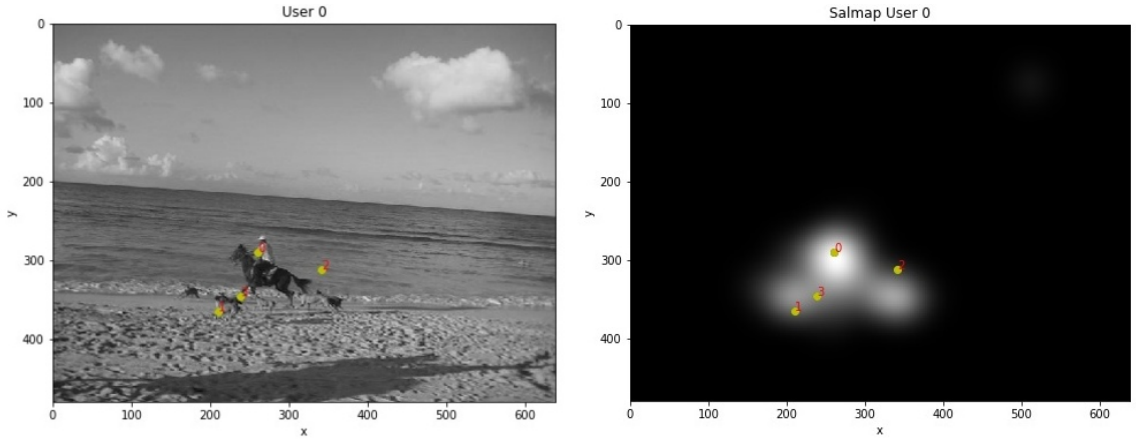

Figure 3.2: Ordered Ground-truth Fixation Points for an observer in the given image.

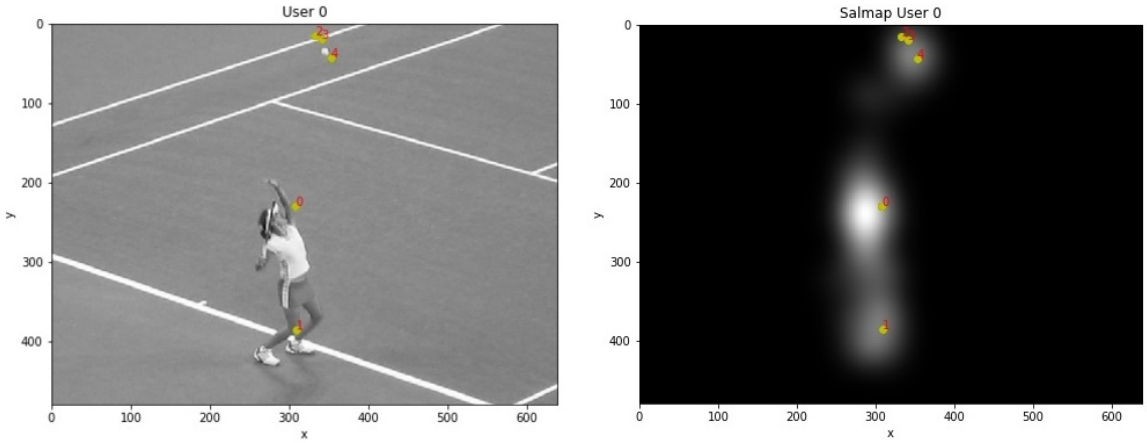

Figure 3.3: Ordered Ground-truth Fixation Points for an observer in the given image.

If you have a look to Figures 3.2, 3.3, 3.4, 3.5, you can see some examples of the images used for testing. Note that when fixation points are plotted in the saliency maps, those saliency maps have been generated merging fixation points from all observers and then blurred using a Gaussian kernel, while fixation points plotted are from a single observer to simplify the visualization and provide a better understanding of what is going on.
From a simple observation, we can conclude that fixation points are highly likely to be provided in order. From the already commented paper [25], we know that first fixation points are almost every time close to the center due to the *center bias* present in human fixations. Fixation points that follow are usually in the most salient area close to the first fixation point. Later ones can be found in less salient regions, sometimes far from the first area observed. These facts can be observed if you give a closer look at the mentioned figures.

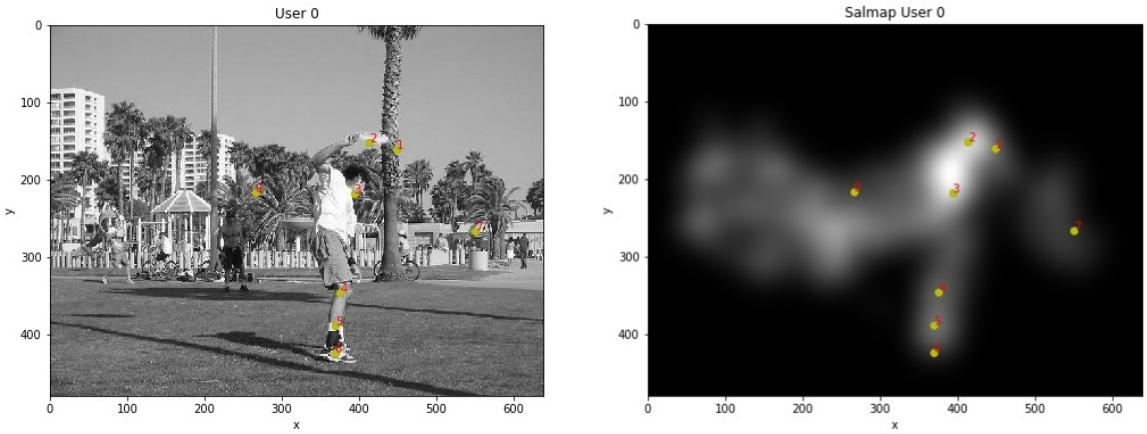

Figure 3.4: Ordered Ground-truth Fixation Points for an observer in the given image.

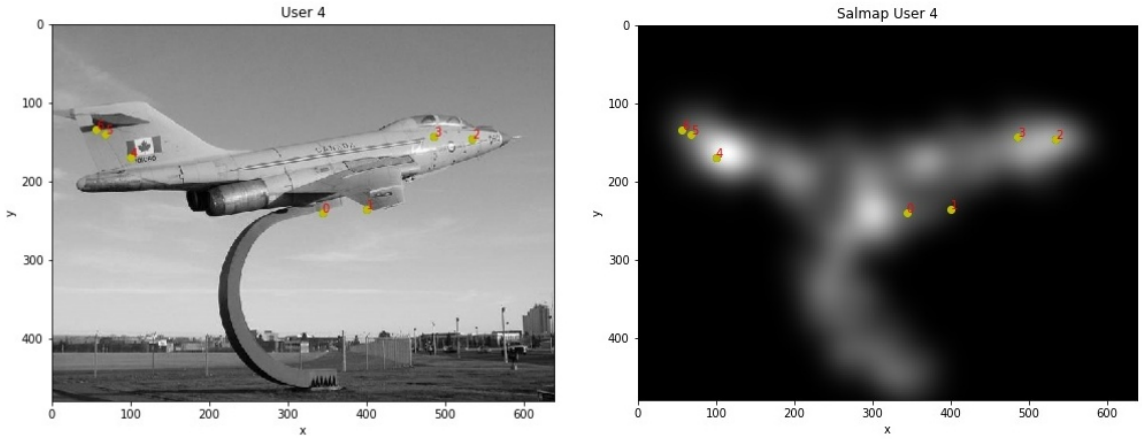

Figure 3.5: Ordered Ground-truth Fixation Points for an observer in the given image.

Chapter 4

Temporally Weighted Saliency Prediction

4.1 Temporally Weighted Saliency Maps

The proper way of generating the Temporally Weighted Saliency Maps (WSMs) is a topic of discussion by itself. We think that the best way would be to weight each fixation point based on their probability of not being randomly selected during the viewing process. The reason behind this is that we consider that by giving less weight to random-like fixation points we would be adding less noise to the system, facilitating its learning process.
Since we needed a baseline WSM to perform our experiments, we have based our criterion for weighting the maps on the paper[37]. In this paper, they point out that earlier fixations are the ones placed in the most salient areas of an image and have a greater consistency between different observers. Once the most salient locations have been visited, there are many less salient selectable positions which are chosen randomly, decreasing the consistency between observers as viewing progresses. Consequently, giving more weight to earlier fixations than later ones would be a way of discerning between which ones are selected randomly and which ones are not.
The way of discerning in a generalized way for all images, which fixation points are going to be considered random-like and as a consequence which points will be assigned a lower weight, has been inspired in the graph from Figure 4.1. In this graph what you can observe is the consistency in selecting the same fixation locations, in function of the fixation number, between different users. As you can see, early fixations are more consistent between different users compared to later ones (note that for Kullback-Leiber divergence [33], a score of zero indicates that we can expect similar, if not the same, behaviour of two different probability distributions. For this specific case, the lower the score, greater the consistency).
Trying to approximate the graph's curve behaviour, we decided to use the decreasing exponential function seen in Eq.4.1 as our weighting function (see Figure 4.2). The first fixation point will be the one with the most amount of weight and it will decrease exponentially among the following fixations.

$$y = e^{-params \cdot x} \tag{4.1}$$

4.1.1 Finding a parameter for the weighting function

As it can observed in the weighting function 4.1, we have a parameter $params$ that determines how quick the curve decreases. Observe Figures 4.3 and 4.4 for a better visualization of what could happen with different values of this parameter. Comparing the Normal Saliency Map with the Maps after weighting we can observe how for small values of the parameter we can not see a real effect on the final map but once this value is increased significantly, we end up having a salient region in the center. This effect is common among all images due to the center bias. Owing to center bias, the first fixation point is highly likely to be in the center of the image.
The goal when choosing a parameter that would fit almost all images, in a generalized way, was

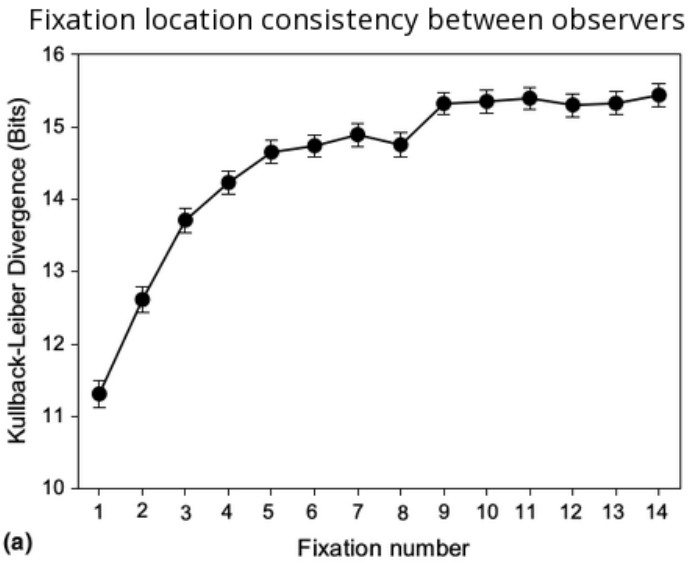

Figure 4.1: Mean Kullback–Leiber divergence (KLdiv) in fixation locations between observers as a function of fixation number. Fixation location consistency between observers is highest for the first fixation and decreases over the course of several fixations on a scene [37]

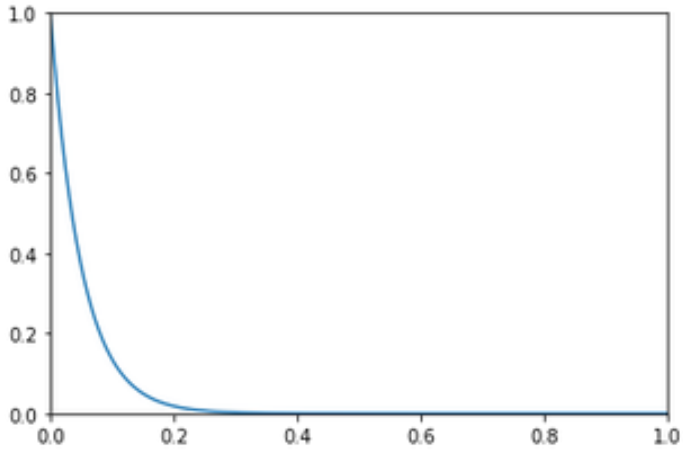

Figure 4.2: Weighting function: $y = e^{-params \cdot x}$

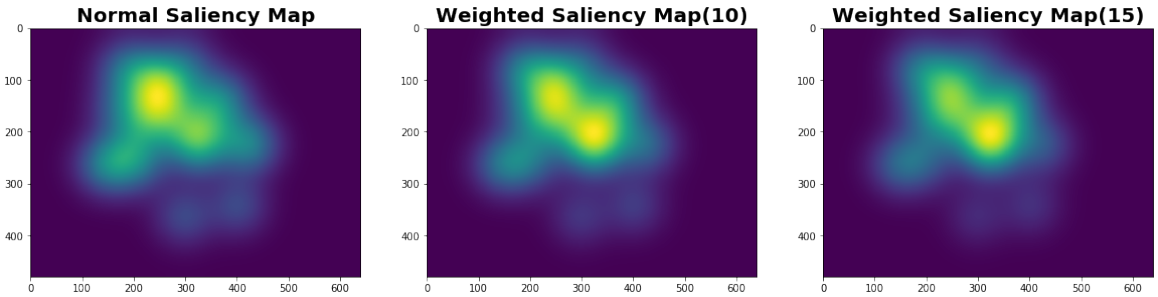

Figure 4.3: Temporally Weighted Saliency Maps for different values of the parameter *params* in the weighting function.

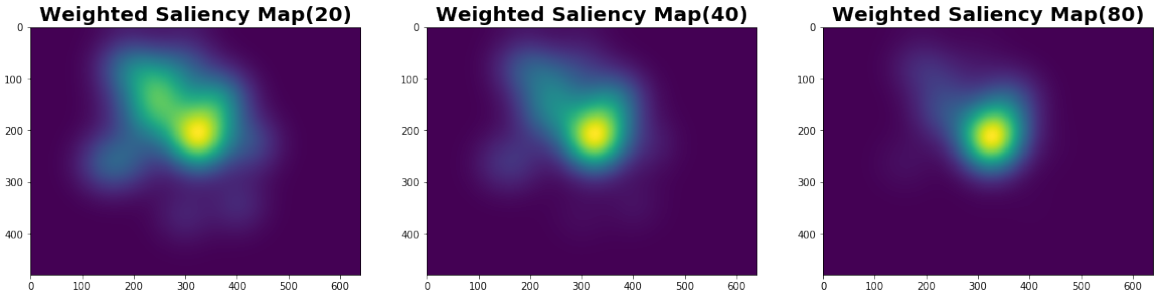

Figure 4.4: Temporally Weighted Saliency Maps for different values of the parameter *params* in the weighting function.

to reproduce the consistency curve seen in the graph from Figure 4.1. With this purpose in mind, we chose a parameter that would give almost no weight after the eighth fixation point and the first four fixation points would be the most relevant. A different value accomplished this goal for the different datasets we worked with, due to SALICON generally having more fixation points across all users than iSUN, we chose a value of 25 for iSUN and 40 for SALICON.

4.2 Models

As we have previously seen in Section 2.2.1, MLNet is a visual attention model with outstanding results in the MIT saliency benchmark [12][8]. It was by far, the best option when we had to chose a model for our experiments. The code is publicly available at *github* [1], its organized and easy to understand, and written in the familiar programming languages and frameworks, Python and Keras. Moreover, in their *github* page, authors claimed to have used the same parameters to produce their results as the ones found in the code published.

Three different versions of MLNet have been used during the development of this project. In this section, we are going to define each of them to facilitate the understanding of our work.

- **MLNet:** When we talk about MLNet, we are referring to the model when it was trained by the authors using the 2015 version of the SALICON dataset. When this version is used in

[1] Code can be found in `https://github.com/marcellacornia/mlnet`

our work, the published MLNet already-trained weights are used and there have not been any modifications to the model provided by the authors.

- **nMLNet:** When we talk about nMLNet, we are referring to the version of MLNet that was trained by us, using Normal Saliency Maps as ground-truth. We generated those maps from ground truth fixation points given in the 2015 version of the SALICON dataset. Ideally, this version should give similar, or the same, results as the version *MLNet* since no parameter was changed with respect to the published code. Unfortunately, as explained in section 4.2.1, we could not replicate the original model results. In any case, we adopted nMLNet as our baseline to be compared with the temporally weighted saliency maps.
- **wMLNet:** When we talk about wMLNet, we are referring to the version of MLNet that was trained by us, using the Weighted Saliency Maps as ground-truth generated for the 2015 version of SALICON dataset.

4.2.1 Replicating MLNet's results

Before moving to training the model using WSMs, we had to try to replicate MLNet original results to have a solid baseline for our experiments. Due to the MLNet architecture being based on the well-known VGG model[36], weights are initialized using the pre-trained weights from VGG-16 as it is done in the original model.
MLNet published weights, were obtained while training using the 2015 version of SALICON dataset[17], therefore we used the same dataset version for this task. Ground-truth Saliency Maps were generated from the dataset fixation points. We did this to test if the way we were creating the maps was done the right way and to apply any necessary changes to the weighted maps we had generated. The model parameters value, were set to the ones that can be found in the model's paper [12] and in the published code at *github*; a batch size of N=10 and SGD is applied with Nesterov momentum of 0.9, weight decay of 0.0005 and a learning rate of 10^{-3}.
After training, predicted saliency maps were computed for the SALICON validation set since the test set is used to evaluate results in the benchmarks, therefore ground-truth fixation points are not published. Predicted maps were then evaluated using the AUC Judd metric commented in Section 4.3.1. Results were compared to the ones obtained by the MLNet authors in *CodaLab*[2]. If you have a look to the Table 4.1), you can observe that we could not reproduce the same results. To make sure that the issue was not our generated ground-truth Normal Saliency Maps, we tried training the model but using the ground-truth saliency maps provided in the dataset instead. We got exactly the same score. For these reasons, after insisting on it for quite a long time, we concluded that the results obtained are the only reproducible results for the published information.

SALICON 2015 Validation set	**AUC Judd**
MLNet published results	0.886
Reproducible MLNet results (baseline)	0.814

Table 4.1: Reproducing MLNet results for the 2015 version of SALICON's validation set

[2]SALICON Challenge 2015: `https://competitions.codalab.org/competitions/3791?secret_key=f8de41aa-090f-4fd1-967e-56fc52ad8456#results`

4.3 Saliency prediction metrics

Saliency prediction field has different metrics for the evaluation of models' predicted maps. You can see some of them if you have a look at Table 4.2. Different metrics require different ground-truth representations. As you can observe in the mentioned table, metrics can be classified into two main groups in function of the ground-truth representation required. Location-based metrics interpret ground-truth maps as binary matrices where only the fixation points positions have high values. Distribution-based interpret ground-truth maps like continuous distributions. In this section, we are going to comment on the metrics used in this work.

Metrics	**Location-based**	**Distribution-based**
Similarity	AUC Judd, sAUC, NSS, IG	SIM, CC
Dissimilarity		EMD, KLdiv

Table 4.2: Saliency Prediction evaluation metrics classified in location-based, and distribution-based. Location-based metrics require discrete Fixation Maps as ground-truth while distribution-based metrics treat both ground truth maps and evaluated saliency maps as continuous distributions. Good scores are those with high values for similarity metrics and low values for dissimilarity ones[9].

4.3.1 Area Under ROC Curve (AUC)

Area Under ROC Curve (AUC) is the most used location-based metric when it comes to the evaluation of saliency maps. This metric treats saliency maps as binary classifiers of fixation points. The Receiver Operating Characteristic (ROC) is used to measure the true and false positive rates for each binary classifier. There are different implementations of this metric that differ in how to calculate true and false positives. In our work, we used the AUC Judd implementation [18].

AUC Judd uses a threshold to determine that all values on the evaluated saliency map above that threshold at fixated pixels will be *true positives*. *False positives* will be all values above the threshold at unfixated pixels. True positive rate (TP rate) and false positive rate (FP rate) are calculated. When the ROC curve can be drawn, the AUC is calculated. An ideal score would be equal to one, while random classifiers score around 0.5 [33].

4.3.2 Kullback-Leibler Divergence (KLdiv)

Kullback-Leibler Divergence (KLdiv) is a commonly used distribution-based metric which calculates dissimilarity between two probability distributions. For this reason, maps are normalized as probability distributions as in Eq.4.2. The resulting score is a measure of the information lost when the predicted saliency map's probability distribution is used to approximate the ground-truth saliency map's probability distribution[33]. KLdiv highly penalizes on mis-detections. As KLdiv measures dissimilarity, a score of zero would mean that both distributions are equal.

$$SM(x) = \frac{SM(x)}{\sum_{x=1}^{X} SM(x) + \epsilon} \tag{4.2}$$

4.3.3 Pearson correlation coefficient (CC)

The Pearson Correlation Coefficient (CC) is a statistical method that when applied to saliency prediction, interprets saliency maps as random variables to measure how correlated or dependent they are from each other. This distribution-based metric penalizes false positives and false negatives equally. The output range for this metric scores is between -1 and 1. Those locations where the evaluated map and the ground-truth map have values of similar magnitudes will give high positive CC values. Scores close to -1 or 1 show an almost perfect linear relationship between both maps [9][33].

Chapter 5

Experiments

5.1 Saliency prediction

5.1.1 Experiment to determine if saliency models have difficulties predicting later fixations

After generating the WSMs as detailed in section 4.1, the first experiment we did was to test our hypothesis that for some images, especially those with few regions of interest, visual attention models have difficulties to predict later fixations.

In order to test the hypothesis, we used MLNet's already trained weights published to compute the predicted maps for the iSUN training set in MLNet. Resulting predicted maps were then evaluated two times, using the evaluation metric called Kullback-Leibler Divergence (KLdiv). The first evaluation was made using the commonly used Saliency Maps (sometimes we refer to them as Normal Saliency Maps, NSM) as ground-truth maps. The second evaluation was made using the Temporally Weighted Saliency Maps (WSM) as ground-truth.
Both evaluations scores were then compared to find which images scored better when they were evaluated using the weighted maps. Having predicted maps that score better when evaluated using WSM rather than NSM, would mean that for the corresponding input images, the model has been able to predict early fixations while having difficulties to predict later ones. A simple observation of the images and their MLNet's predicted maps that had significantly better scores for the evaluation with the WSMs, showed evidence enough to prove our hypothesis.
Results can be seen in Section 5.1.3.1.

5.1.1.1 Choice of a proper evaluation metric

As we have previously seen in Section 4.3, there are several metrics to evaluate saliency maps predicted by a model. Metrics can be categorized as location-based or distribution-based. The main difference is in terms of the input, whether they require discrete Fixation Maps as ground-truth or continuous maps[9]. Since the objective of the experiment was evaluating predicted maps using NSM and WSM, and the weighted maps from their nature cannot be discretized, we had to choose from the distribution-based metrics.
We chose to evaluate on Kullback-Leibler Divergence (KLdiv), since this metric highly penalizes on mis-detections, therefore it suits the purpose of this experiment, which consists in evaluating if in some cases visual attention models have difficulties learning to predict later fixation points. Note that since KLdiv measures dissimilarity, a score of zero would mean that both distributions are equal, the bigger the score value, greater the dissimilarity.

5.1.2 Study on the effect of Weighted Saliency Maps on a visual attention model's performance

Once we had evidence that in some cases visual attention models have difficulties in predicting later fixation points, we had to test if training using Weighted Saliency Maps (WSM) could improve model's performance. The hypothesis was that treating all fixation points equally when its known that later fixation points can be selected randomly, could be adding an unnecessary noise to the input of the system that could obstruct the learning process. For these reasons, we decided to train two versions of MLNet. The first one using Normal Saliency Maps (NSMs) as ground-truth during training (nMLNet), and the second one using WSMs (wMLNet). Observing an improvement when evaluating both models using NSMs as ground-truth during the evaluation, would prove a rise in the model's performance.

5.1.2.1 Evaluation

Predicted maps obtained using wMLNet, and the ones obtained when using nMLNet, were then evaluated using three different metrics, Kullback-Leibler Divergence (KLdiv), Pearson's Correlation Coefficient(CC) and AUC Judd, where AUC stands for Area Under ROC Curve. All metrics were evaluated using NSMs as ground-truth and for the distribution-based metrics, KLdiv and CC we also evaluated results using WSMs. When evaluating using AUC Judd, since it is a location-based metric which expects a discrete map as ground-truth, we could only evaluate using NSMs, in this case, NSMs were discrete maps, also called Fixation Maps. Fixation Maps are binary matrices where only the positions of fixation points have a high value.
To evaluate any improvement in performance, we compared the evaluation scores for the predicted maps of both versions of the model. Results showed an improvement for all metrics in wMLNet compared to nMLNet, therefore a performance boost was confirmed. A deeper insight of results obtained can be seen in Section 5.1.3.2.

5.1.3 Results

5.1.3.1 Determining if saliency models have difficulties predicting later fixations

As explained in Section 5.1.1, the purpose of this experiment was to determine if, for some images, especially those with few regions of interest, visual attention models have difficulties to predict later fixation points. For these reasons, we used the Kullback-Leibler Divergence (KLdiv) metric to evaluate MLNet's predicted maps for the iSUN training set. Remember that KLdiv penalizes mis-detections, therefore, is the most suitable metric for the purpose of our observation.

Figure 5.1, shows a histogram of the KLdiv scores when MLNet's predicted maps, for all images in the iSUN training set, were evaluated using Normal Saliency Maps (NSMs) as ground-truth. Note that, as previously mentioned, for KLdiv metric, the lowest the score value, better the similarity between the evaluated map and the ground-truth map. If we compare this histogram with the histogram seen in Figure 5.2, that shows the scores for the same predicted maps when those have been evaluated using Weighted Saliency Maps (WSMs), they look very similar at first sight. The only noticeable difference can be appreciated especially in the score range between

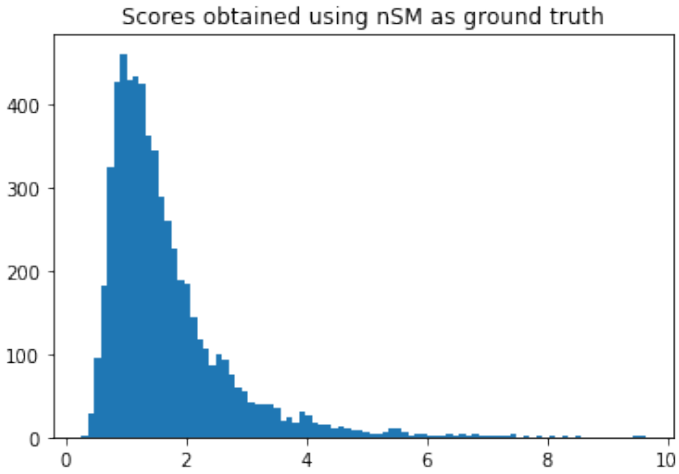

Figure 5.1: Histogram of the KLdiv scores for all images when the evaluation was made using NSMs as ground-truth

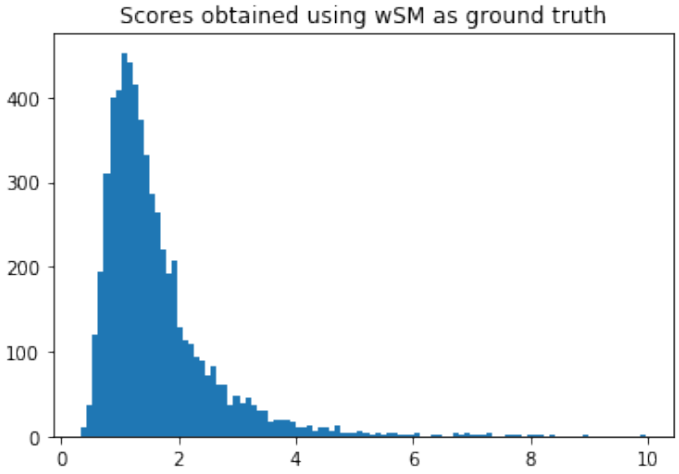

Figure 5.2: Histogram of the KLdiv scores for all images when the evaluation was made using WSMs as ground-truth

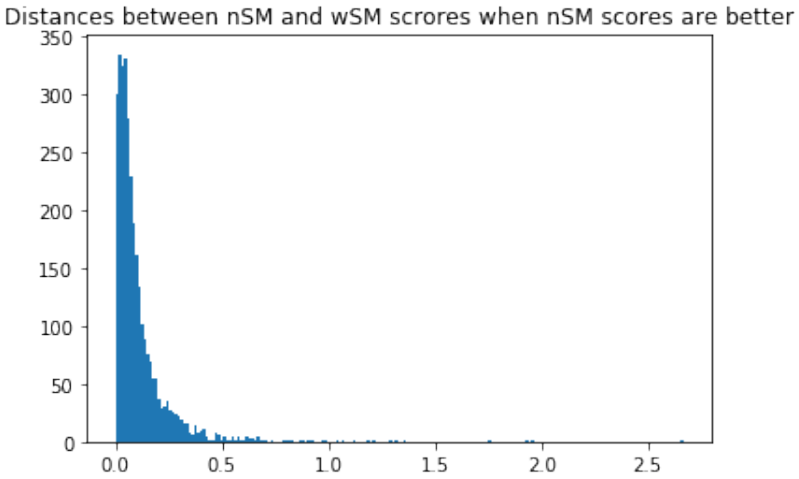

Figure 5.3: Distances between KLdiv scores evaluated with NSMs and WSMs for images that scored better when the evaluation was made using NSMs as ground-truth

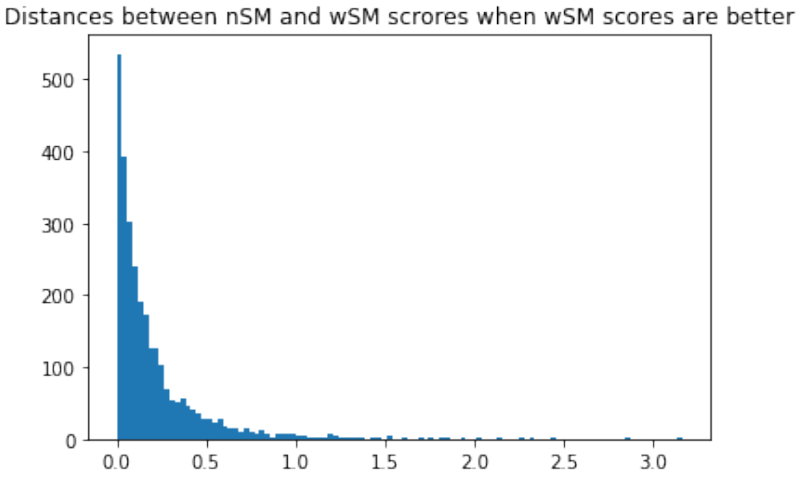

Figure 5.4: Distances between KLdiv scores evaluated with NSMs and WSMs for images that scored better when the evaluation was made using WSMs as ground-truth

4 and six, where we can observe more maps with high scores, worse scores, when maps have been evaluated using NSMs as ground-truth, compared to when maps have been evaluated using WSMs instead. This difference can be easily appreciated in the histograms from Figures 5.3 and 5.4, which shows the distances between the scores obtained for the same image when evaluating using NSMs compared to when evaluating using WSMs. Figure 5.3 shows these distances for predicted maps that scored better when evaluating using NSMs as ground-truth and Figure 5.4, when they scored better when evaluated using WSMs instead. Almost half of the evaluated maps scored better for WSMs.

At first sight, we can easily see that there are more maps that scored significantly better when evaluating with WSMs compared to the ones that scored much better for NSMs.

Notice that in most cases scores for both evaluations were almost the same. Specifically, 90.35% of maps that scored better when evaluated using NSMs have a distance of less than 0.25 to the WSM score, and 74.73% for the case of maps that scored better when evaluating using WSMs. Since we wanted to evaluate cases where scores were significantly different, we are going to focus on these to draw our conclusions.

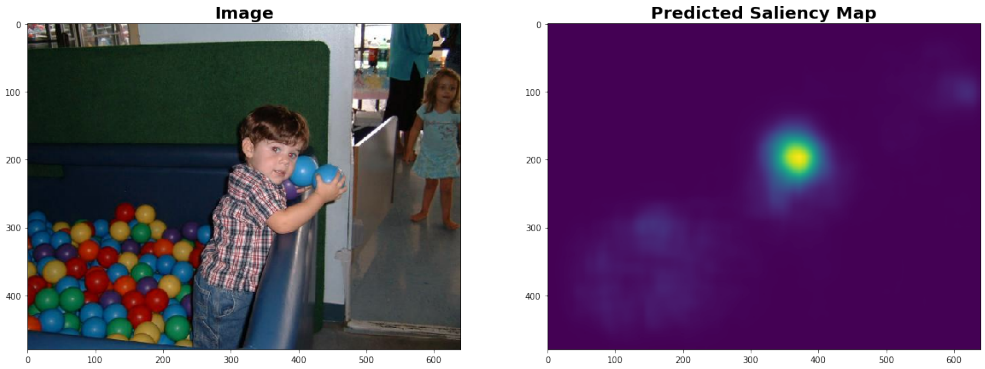

Figure 5.5: Example of an image and it's MLNet predicted map, that scored significantly better when the evaluation was made using WSMs as ground-truth rather than when NSMs were used

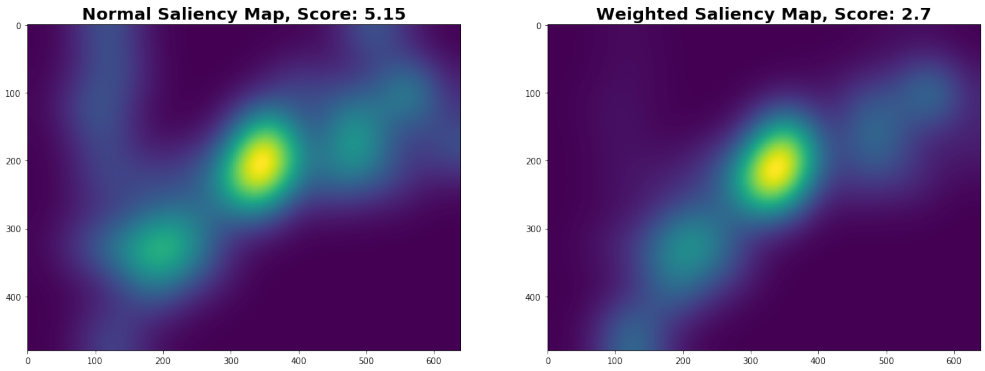

Figure 5.6: Evaluation for both kinds of ground-truth for the MLNet's predicted Map from Figure 5.5

We decided to plot some of the images that had a greater distance between scores obtained when evaluating with both kinds of ground-truths to try to find a pattern that explained this difference. Figures 5.5 and 5.7 are examples of images that scored significantly better when evaluating using WSMs. You can also observe the ground-truth maps for these images and their scores when used for the evaluation at Figures 5.6 and 5.8 respectively. We noticed that what all those images had in common is that they were images with few regions of interest and the model had learned to predict only early fixation points. Observe how in both examples, predicted maps are more similar to the WSM ground-truth than to the NSM. Since WSMs have more weight to early fixation points, it confirms that the model has learned to predict those. Consequently, the hypothesis treated in the experiment was confirmed by this observation.

From the observation performed on images that scored significantly better when the evaluation was made using NSMs as ground-truth, we concluded that scores were bad for both maps, so we could not extract any further conclusions. See some examples in Figures 5.9, 5.10, 5.11 and 5.12.

This Section results encouraged us to continue studying the effect that Temporally Weighted Maps could have to the Saliency Prediction Field.

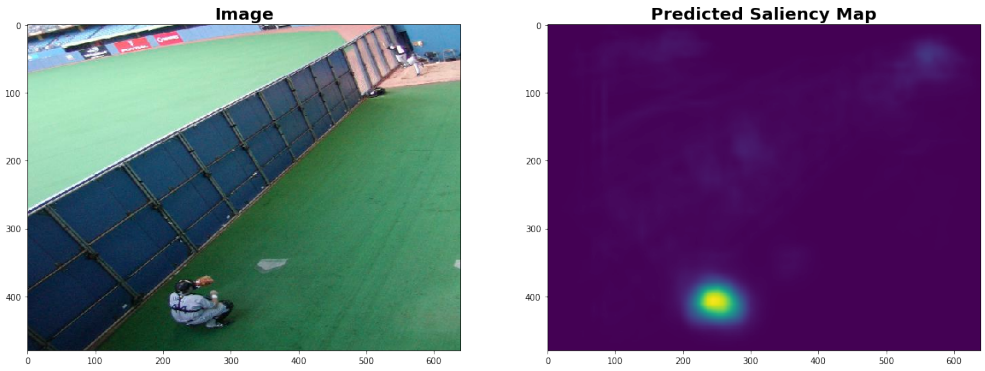

Figure 5.7: Example of an image and it's MLNet predicted map, that scored significantly better when the evaluation was made using WSMs as ground-truth rather than when NSMs were used

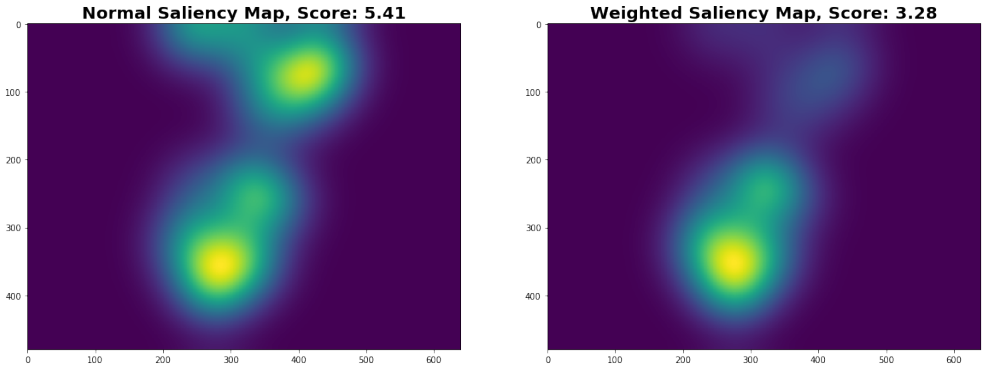

Figure 5.8: Evaluation for both kinds of ground-truth for the MLNet's predicted Map from Figure 5.7

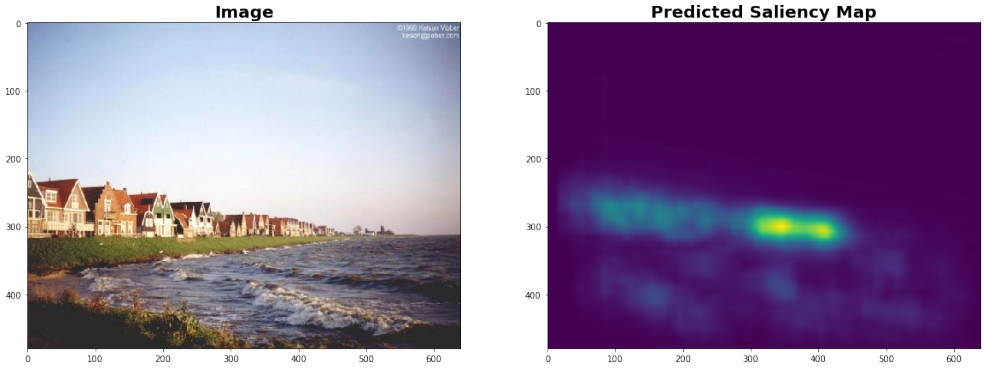

Figure 5.9: Example of an image and it's MLNet predicted map, that scored significantly better when the evaluation was made using NSMs as ground-truth rather than when WSMs were used

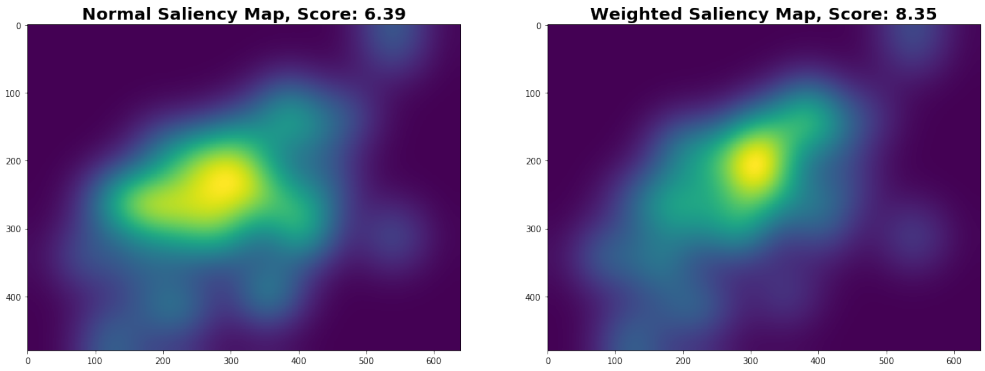

Figure 5.10: Evaluation for both kinds of ground-truth for the MLNet's predicted Map from Figure 5.9

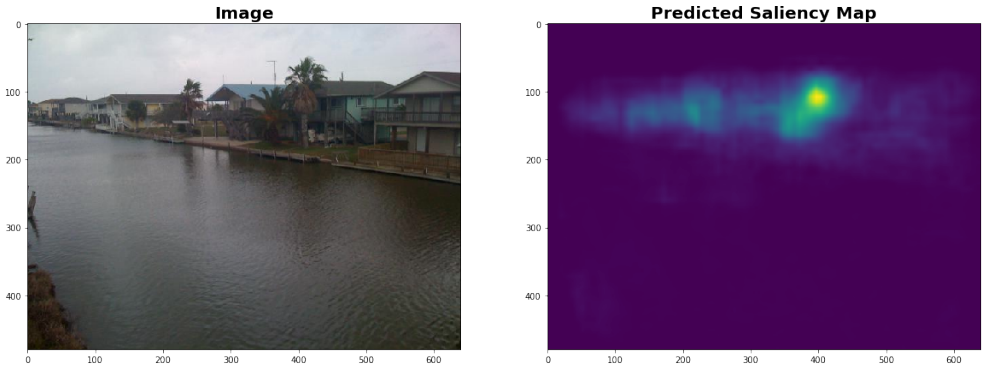

Figure 5.11: Example of an image and it's MLNet predicted map, that scored significantly better when the evaluation was made using NSMs as ground-truth rather than when WSMs were used

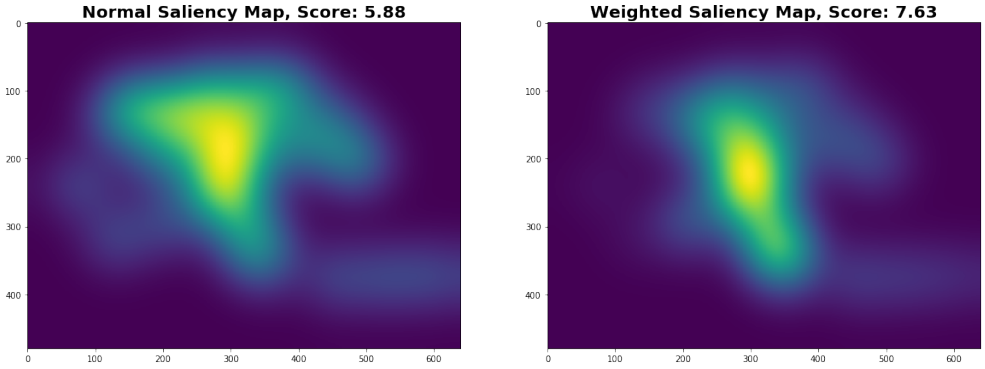

Figure 5.12: Evaluation for both kinds of ground-truth for the MLNet's predicted Map from Figure 5.11

5.1.3.2 Study on the effect on the use of Weighted Saliency Maps on a visual attention model's performance

As detailed in Section 5.1.2, the experiment consisted on training two versions of MLNet, the first one nMLNet, using Normal Saliency Maps (NSMs) as ground-truth and the second one wMLNet, using Temporally Weighted Saliency Maps(WSMs) instead. The purpose of the experiment was to test if training using this weighted maps could improve visual attention models' performance, confirming the hypothesis that adding less weight to fixation points that are selected randomly during visualization, later ones as seen in [37], would be adding less noise to the input of the model, facilitating the learning process.

Predicted maps from nMLNet and predicted maps from wMLNet, were then evaluated using two distribution-based metrics, Kullback-Leibler Divergence (KLdiv) and Pearson's Correlation Coefficient(CC), and a location-based metric, AUC Judd, where AUC stands for Area Under ROC Curve. For this last metric as previously commented, we could only evaluate results for NSMs as ground-truth maps have to be discrete Fixation Maps, therefore, WSMs cannot be used. Observe results obtained for the different metrics: AUC Judd in Table 5.1, KLdiv in Table 5.2 and finally CC in Table 5.3.

Ground-truth	**nMLNET**	**wMLNet**
Normal Saliency Maps	0.814	0.816

Table 5.1: Evaluation results for AUC Judd. The higher the score, the better the similarity between the evaluated map and the ground-truth

Ground-truth	**nMLNET**	**wMLNet**
Normal Saliency Maps	1.332	1.039
Weighted Saliency Maps	1.493	1.136

Table 5.2: Evaluation results for Kullback-Leibler Divergence (KLdiv). The lower the score, the better the similarity between the evaluated map and the ground-truth

Ground-truth	**nMLNET**	**wMLNet**
Normal Saliency Maps	0.534	0.539
Weighted Saliency Maps	0.509	0.517

Table 5.3: Evaluation results for Pearson's Correlation Coefficient(CC). The higher the score, the better the similarity between the evaluated map and the ground-truth

Comparing evaluation scores between nMLNet and wMLNET, we can observe that results are better when the model was trained using Weighted Saliency Maps(WSMs). As we can see, wML-Net scores are better than nMLNet when predicted maps were evaluated using Normal Saliency Maps as ground-truth (conventional way of evaluating) and also when they were evaluated using WSMs instead. This fact demonstrates an improvement on model's performance, proving our hypothesis true.
Even though the metrics scores have improved, which makes us think we are on the right track, differences are not as big as we expected them to be. We think that the main reason behind this is the way of generalizing for all images the weighting applied when creating the WSMs. This method was good for a baseline map, but we think that it can be significantly improved. When

we studied the dataset, we noticed that each image has a different number of fixation points and that this number can also vary for different observers. For this reason, we think that a way that treats images independently needs to be applied when properly weighting the maps. Some ideas we have in this regard are commented in Section 8.1.

From these results we, also, see the need for a metric able to understand Weighted Saliency Maps. Since we know that models have difficulties predicting random-like fixation points, we think that results should be evaluated taking this into account. From our point of view a proper way of evaluating models results using Weighted Saliency Maps as a ground-truth, would imply a metric that should produce greater scores if the model has correctly learned to predict those regions with a greater weight on the ground-truth map, or hard penalizations if the model misses those predictions, and should give low penalties for mis-predicted regions that had low weigh on the ground-truth map.

5.2 Visual search

From our satisfactory results when improving saliency models' performance using Temporally Weighted Saliency Maps, we thought that testing their performance in different computer vision tasks could be an interesting addition to our work. A collaboration with Eva Mohedano allowed us to explore this option. We tested our weighted maps on the bag of words model called SalBow[29], which was designed to address the instance search task.

In recent years, visual content has become part of our lives, being the most shared content type in social media. The increasing amount of visual data lead to the need for systems able to automatically retrieve images based on their content. Instance search is the task that addresses the problem of retrieving images from a database that contain an instance of a query[35]

5.2.1 Experiment

SalBoW is a retrieval framework based on bags of local convolutional features (BLCF). SalBow builds an efficient image representation using saliency maps to weight the contribution of local convolutional representations for the instance search task. This approach outperformed the state-of-the-art on the INSTRE benchmark for image retrieval, without the need of fine-tuning features or conducting region analysis or spatial verification.

As you can see in Figure 5.13, the model is divided in two parts. The first module extracts semantic features from a Convolutional Neural Network and using the K-means algorithm on the extracted features, a visual vocabulary is learned. The outcome of this procedure is an *assignment map*, a semantic representation for each image. This representation's main advantage, is that the spatial layout of the image is preserved; it is possible to apply a spatial weighting scheme before the construction of the final Bag of Words (BoW) representation. For the weighting scheme, saliency maps are down-sampled to match the spatial resolution of assignment maps and normalized to the range between 0 and 1 to be used as weights. The final BoW representation is a histogram where each component is the sum of the spatial weight assigned to a particular word.

On the experiment performed, Weighted Saliency Maps predicted by wMLNet for the INSTRE dataset were used for the saliency weighting. SalBoW performance was evaluated using the Mean

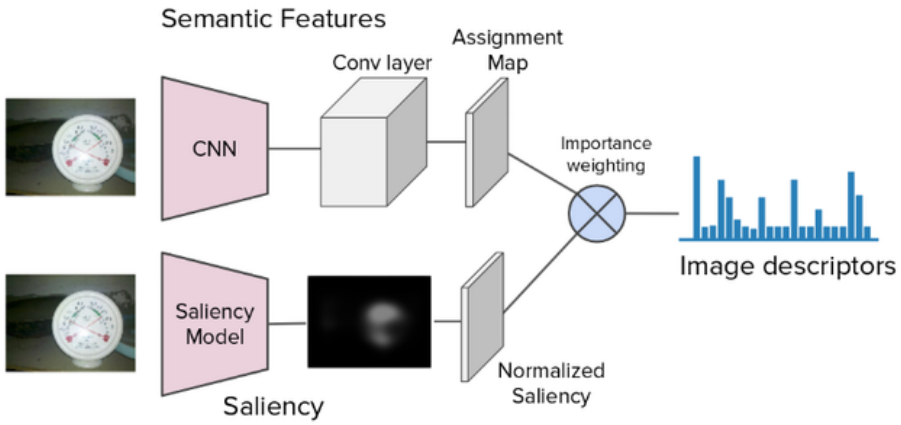

Figure 5.13: The Bag of Local Convolutional Framework (BLCF) pipeline with saliency weighting [29]

.

Average Precision (mAP) metric and results were compared with the performance results when Normal Saliency Maps produced by nMLNet, were used for the weighting instead.

5.2.2 Results

SalBow performance was evaluated using the mean Average Precision (mAP) metric. Results for the evaluation when the weighting was done using Weighted Saliency Maps produced by wMLNet and when the weighting was done using Normal Saliency Maps predicted by nMLNet instead, can be seen in Table 5.4.

Saliency Maps type	**mAP**
Normal Saliency Maps	0.6730
Weighted Saliency Maps	0.6743

Table 5.4: Evaluation of SalBoW performance with the Mean Average Precision metric

As you can observe in the aforementioned table, results show an improvement when Weighted Saliency Maps were used for the weighting purpose. Even though results do not achieve the state-of-the-art, due to the complications on replicating MLNet published results and the fact that Weighted Saliency Maps' weighting can be significantly improved, we find these results encouraging and it strengthens our belief that further research should be done in our line of work.

Chapter 6

Ethics

In recent years, the field of deep learning, also called Artificial Intelligence (AI), has been growing adepts exponentially due to its potential that can be exploited in areas that go from health-care to security or automation. Due to the increasing number of applications for this technology, it has recently caught the media attention. Discussions have mainly been focused on the fact that AI has grown and is still growing so fast that there is still a lack of regulation and ethics code that could consider the social effects of this technology. As mentioned, AI has spread to many fields, and each one has its own ethical concerns. For these reasons, we are going to give a bigger picture of its possible impacts on society.

The main problem with AI systems is that results are always going to be a reflection of the datasets used to train the models with. For instance, in [21], they demonstrated that facial-recognition systems, consistently have lower matching accuracies for females, Afro-Americans and the age group 18–30. They conclude that the cause is that datasets are not evenly distributed taking demographics into account. Imagine the possible consequences in peoples lives when these systems are used in intelligence, law enforcement or especially in the health-care field, where a mis-prediction could lead to a wrong diagnose. Sometimes the problem is not on the accuracy of the predictions for some isolated groups, but a biased dataset that propagates its biases to the system results. Picture a system used in policing to find suspects based on their profile. If the system was trained using a dataset based on historical data, this system would be perpetuating prejudices that existed in the past towards its results[1]. Solving this kind of issues imply more than tweaking the numbers to try to remove systemic inequalities and biases. In many cases, only engineers and computer scientists are involved in the process of building this kind of systems. We consider that experts from the area being treated should be part of the process from the beginning to provide insights on the ethical challenges on the field and what could be done to make the systems fairer to them. When the problem being faced is the accuracy for some groups of the society, as presented in [21] for the case of face-recognition systems, we agree with the solution proposed in the paper, where they suggest using different algorithms, each one trained for a specific group. This process should lead to an improved accuracy especially for the groups affected. An alternative solution would be training using evenly distributed datasets. But when we talk about large datasets, this can be a big challenge itself.

We are currently living in a world where data is extracted from almost everything we do that involves the use of an electronic device. From our scrolling pattern on social media to our profile information. AI systems are a powerful tool that can use all kinds of data to fulfil the purpose they are designed for. For this reason, there is an urgent need for regulation on the use of the data collected. A recent example, the Facebook case with Cambridge Analytica that as investigations suggest, Facebook users data was being used to develop political propaganda campaigns, can be a perfect example of what can happen when the data is in the wrong hands.

If we move to the sector of automation, the discussion being held is the same as for the technology field in general. The main concern is the jobs that may disappear because of AI. An

[1] Example inspired in the interview from `https://www.wired.com/story/why-ai-is-still-waiting-for-its-ethics-transplant/`

example of it could be the driver-less cars and lorries from Uber that in a near future might take the roads. It cannot be denied that whereas many engineering and computer science jobs will be created, automation will directly affect many middle-class jobs in a near future. But since we are talking about systems that can directly affect positively the economy and improve aspects of everyday life, politicians will have to intervene to make sure everyone can benefit, for instance promoting policies that help with the creation of new jobs for those affected.

In the specific field that involves this project, the saliency prediction field, we could not find any direct ethical concerns. Even so, saliency prediction is usually applied to improve other computer vision tasks. So we consider that it might indirectly affect the ethical concerns previously explored.

The potential health-risks for this project, are the ones developed from spending too much time in front of a computer. This prolonged behaviour could eventually lead to what is called Computer Vision Syndrome (CVS) which causes symptoms that include eye strain, tired eyes, irritation, redness, blurred vision, and double vision. Experts recommend a proper lighting when working on the computer, a proper position with respect to the monitor and regular work breaks[7]. In the same line, being seated during too much time can affect our posture. Dealing with this problem requires proper education on how to sit to minimize these risks and taking breaks to walk.

Chapter 7

Budget

This thesis has been developed using the computing resources provided by Image Processing Group of UPC; there are no hardware costs. However, we can use *Amazon Web Services* (AWS) to approximate the real cost if computational resources would not have been provided by the university.

The resources used on this thesis were of 1 GPU with 11GB of GDDR SDRAM (Graphics Double Data Rate Synchronous Dynamic RAM) and about 30GB of regular RAM. The most similar resources can be found in EC2 instance; *p2.xlarge*. This service provides 1 GPU, 12GB GDDR SDRAM and 1 CPU with 61GB of RAM [1]. The cost is 0,9$ per hour; 21,6$ a day, equivalent to 18,3€(1€equals 1,18$ on 9/06/2018). For about 90 days of usage, the total cost ascents to 1647€.

The software used was open-source; there are no costs associated. Since this is a comparative study we are not considering any maintenance cost. The only relevant costs for the thesis comes from the salary of the researchers for the 17 weeks duration of the project, as depicted in the Gantt diagram in Figure 1.6. To calculate this costs, I will consider myself an Undergraduate Research Assistant and Senior Engineers for the two professors who have advised me. In addition I've taken into account the salary of a short contribution that a Research Assistant did to the evaluation of our weighted maps for the visual search task. You can see the calculation of the total costs in Table 7.1.

	Amount	**Wage/hour**	**Dedication**	**Total**
Undergraduate Research Assistant	1	8,00 €/h	30 h/week	4080 €
Research Assistant	1	20,00 €/h	8 h	160 €
Senior Engineer	2	40,00 €/h	3 h/week	4080 €
Computational Resources				1647 €
			Total	9967 €

Table 7.1: Budget calculations for the thesis development

Chapter 8

Conclusions

Weighted Saliency Maps were first used to evaluate MLNet's predicted maps for the iSUN training set. The purpose of this experiment was to test the hypothesis, that for some images, especially those with few regions of high salience, visual attention models have difficulties predicting later fixation points. From our observation of those images that had a greater difference between the evaluation scores when they were evaluated using the weighted maps compared to when they were evaluated using the normal Saliency Maps, we concluded that images that scored significantly better when evaluated using Temporally Weighted Saliency Maps, had few relevant regions in them. Since weighted maps have more weight to early fixations compared to later ones, it was confirmed that for these images, the model had difficulties on learning later fixation points. For those images that scored significantly better when the evaluation was performed using the unweighted maps, we found that the resulting scores were unquestionably bad for both evaluations; we could not draw any further conclusions.

Our second experiment was meant to test if visual attention models could improve performance, if Weighted Saliency Maps were used as ground-truth maps during training. We evaluated predicted maps from both versions of the model, training using normal maps as ground-truth (nMLNet) and using weighted maps instead (wMLNet), using three different metrics for Saliency prediction for both kinds of ground-truth. All cases scored better for wMLNet; we concluded that visual attention model's performance can, indeed, be improved by the use of weighted maps. From results not being improved as much as we would have them expected to, we think that even though the approach used for weighting the maps was good for baseline purposes, the weighting can be significantly improved (Our suggestions on this matter can be seen in section 8.1). Besides, our scores could not reach state-of-the-art scores due to the complications found when trying to replicate MLNet's published results that led us to lower our expectations for the model baseline. Even so, we consider our study a success, being that our hypothesis was proved, and we see the need for further studies on this line of research.

Since from our study we concluded that visual attention models have difficulties learning fixation points selected randomly during visualization, it makes us think that model's predictions should not be evaluated as if we were expecting the model to be able to predict those fixations. We consider that a new metric for saliency evaluation should be implemented to take these facts into account. We think that a proper evaluation metric that expects Weighted Saliency Maps as ground-truth, should produce greater scores if the model has correctly learned to predict those regions with a greater weight on the ground-truth map, or hard penalizations if the model misses predicting those positions. In addition, we think that it should give low penalties for mis-predicted regions that had low weigh on the ground-truth map.

Finally, we used our weighted maps in SalBow[29], a model that tackles the instance search task. In this experiment, our maps were used for weighting the contribution of local convolutional representations extracted from a CNN network. Results showed an improvement in model's performance when Temporally Weighted Saliency Maps were used for the weighting instead of when Normal Saliency Maps were used. For these reasons, we conclude that weighted maps should be further studied since they have the potential to improve performance in other computer vision tasks.

8.1 Future work

In this project our baseline Weighted Saliency Map was generated using a weighting function suitable for generalizing across all images. From our results we concluded that the proper way of weighting a map should be able to treat images more individually, for these reasons, we propose two ways to approach this task:

- We have seen that order is an important factor, but the time spend on each fixation point also tells if what the observers are watching attracts their interest. Consequently, we think that fixations with more time spend and early in viewing should have a greater weight than later fixations and those with a small-time spent on them.
  This method could be used as an alternative way of generalization across all images, but it could also be applied individually if the weighing adjusted to the number of fixation points and the time spent in each of them for each image.

- We considered the order as a way of discerning which fixation points are selected randomly when an observer visualizes an image. We think that another way of telling which fixations aren't random-like could be looking for those fixation points that are common between observers for each specific image.
  A kernel with a given bandwidth could be used to tell if two fixation points could be considered the same fixation point. With this strategy, fixation points common between different observers would have a greater weight. The more observers that have visited the same position, the more weight added to the fixation point.
  With this strategy individualization for all images is accomplished, and it seems to us a more reliable way of telling which fixation points should be considered random-like.

Bibliography

[1] Amazon web services `https://aws.amazon.com/es/?nc2=h_lg`.

[2] Convolutional neural networks for visual recognition `http://cs231n.github.io/neural-networks-1/`.

[3] Eric Arazo Sánchez. The impact of visual saliency prediction in image classification. Master's thesis, Universitat Politècnica de Catalunya, 2017.

[4] Moawad Assaad. Back propagation and neural networks explained `https://datathings.com/blog/post/neuralnet/`.

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

[6] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[7] Clayton Blehm, Seema Vishnu, Ashbala Khattak, Shrabanee Mitra, and Richard W Yee. Computer vision syndrome: a review. *Survey of ophthalmology*, 50(3):253–262, 2005.

[8] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark, 2015.

[9] 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.

[10] Zoya Bylinskii, Adrià Recasens, Ali Borji, Aude Oliva, Antonio Torralba, and Frédo Durand. Where should saliency models look next? In *European Conference on Computer Vision*, pages 809–824. Springer, 2016.

[11] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[12] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A deep multi-level network for saliency prediction. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3488–3493. IEEE, 2016.

[13] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[15] T Huang. Computer vision: Evolution and promise. 1996.

[16] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[17] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1072–1080. IEEE, 2015.

[18] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. 2012.

[19] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th international conference on*, pages 2106–2113. IEEE, 2009.

[20] Wolf Kienzle, Matthias O Franz, Bernhard Schölkopf, and Felix A Wichmann. Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of vision*, 9(5):7–7, 2009.

[21] Brendan F Klare, Mark J Burge, Joshua C Klontz, Richard W Vorder Bruegge, and Anil K Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801, 2012.

[22] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[24] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint arXiv:1411.1045*, 2014.

[25] Patrick Le Callet and Ernst Niebur. Visual attention and applications in multimedia technologies. *Proceedings of the IEEE*, 101(9):2058–2067, 2013.

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[27] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[28] Zhaoping Li. A saliency map in primary visual cortex. *Trends in cognitive sciences*, 6(1):9–16, 2002.

[29] Eva Mohedano, Kevin McGuinness, Xavier Giro-i Nieto, and Noel E O'Connor. Saliency weighted convolutional features for instance search. *arXiv preprint arXiv:1711.10795*, 2017.

[30] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

[31] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–606, 2016.

[32] Michael I Posner. Orienting of attention. *Quarterly journal of experimental psychology*, 32(1):3–25, 1980.

[33] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and human fixations: state-of-the-art and study of comparison metrics. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1153–1160. IEEE, 2013.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[35] Amaia Salvador, Xavier Giró-i Nieto, Ferran Marqués, and Shin'ichi Satoh. Faster r-cnn features for instance search. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, pages 394–401. IEEE, 2016.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Benjamin W Tatler, Roland J Baddeley, and Iain D Gilchrist. Visual correlates of fixation selection: Effects of scale and time. *Vision research*, 45(5):643–659, 2005.

[38] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006.

[39] Amazon Mechanical Turk. Amazon mechanical turk. *Retrieved August*, 17:2012, 2012.

[40] Eleonora Vig, Michael Dorr, and David Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2798–2805, 2014.

[41] Lingyun Zhang, Matthew H Tong, Tim K Marks, Honghao Shan, and Garrison W Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32–32, 2008.

[42] Yinda Zhang, Fisher Yu, Shuran Song, Pingmei Xu, Ari Seff, and Jianxiong Xiao. Large-scale scene understanding challenge: Eye tracking saliency estimation.