Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

The Time Dimension of Visual Attention Models

A Degree Thesis
Submitted to the Faculty of the
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

In partial fulfilment of the requirements for the
Degree in Telecomunications Engineering

Author: Marc Assens Reina
Advisors: Xavier Giró-i-Nieto, Kevin McGuiness and Noel O'Connor

Universitat Politècnica de Catalunya (UPC)
2016 - 2017

Abstract

This thesis explores methodologies for scanpath prediction on images using deep learning frameworks.

As a preliminary step, we analyze the characteristics of the data provided by different datasets. We then explore the use of Convolutional Neural Networks (CNN) and Long-Short-Term-Memory (LSTM) newtworks for scanpath prediction. We observe that these models fail due to the high stochastic nature of the data.

With the gained insight, we propose a novel time-aware visual saliency representation named *Saliency Volume*, that averages scanpaths over multiple observers.

Next, we explore the SalNet network and adapt it for saliency volume prediction, and we find several ways of generating scanpaths from saliency volumes.

Finally, we fine-tuned our model for scanpaht prediction on 360-degree images and successfully submitted it to the Salient360! Challenge from ICME. The source code and models are publicly available at `https://github.com/massens/saliency-360salient-2017`.

Resum

Aquesta tesi explora diverses metodologies per a la predicció de *scanpaths* en imatges utilitzant llibreries de *deep learning*.

Com a pas previ, s'analitzen les característiques de les dades proporcionades per diferents bases de dades. A continuació, explorem l'ús de Xarxes Neuronals Convolucionals (CNN) i xarxes Long Short Term Memory (LSTM) per a la predicció de *scanpaths*. Observem que aquests models fracassen a causa de la gran naturalesa estocàstica de les dades.

Amb el coneixement adquirit, proposem una nova representació d'atenció visual que inclou una dimensió temporal anomenada *Volum d'atenció visual*, que fa la mitjana dels *scanpaths* de múltiples observadors.

A continuació, explorem la xarxa de SalNet i l'adaptem per a la predicció de volums d'atenció visual i trobem diferents formes de generar *scanpath* a partir de volums d'atenció visual.

Finalment, hem fet *fine-tunning* del nostre model per a la predicció de scanpaths en imatges de 360 graus i l'hem enviat al Salient360! Challenge. El codi font i els models estan disponibles públicament a `https://github.com/massens/saliency-360salent-2017`.

Resumen

Esta tesis se exploran las diferentes metodologías para la predicción de *scanpaths* en imágenes utilizando librerías de *deep learning*.

Como paso preliminar, analizamos las características de los datos proporcionados por las diferentes bases de datos. A continuación, exploramos el uso de Redes Neuronales Convolucionales (CNN) y de redes Long-Short-Term-Memory (LSTM) para la predicción de *scanpaths*. Estos modelos fallan debido a la alta naturaleza estocástica de los datos.

Con el conocimiento adquirido, proponemos una nueva representación de la atención visual que incluye infomración temporal llamada *Volumen de atención visual*, que promedia los *scanpaths* sobre múltiples observadores.

A continuación, exploramos la red SalNet y la adaptamos para la predicción de volúmenes de atención visual, y encontramos diferentes formas de generar *scanpaths* a partir de volúmenes de atención visual.

Por último, hemos adaptado nuestro modelo para la predicción de *scanpaths* en imágenes de 360 grados y lo hemos enviamos al Salient360! Challenge del ICME. El código fuente y los modelos están disponibles públicamente en `https://github.com/massens/saliency-360salient-2017`.

Acknowledgements

I would like to specially thank my supervisors Xavi Giro and Kevin McGuiness for his effort and help. Xavi has not only helped me with this project, but also has given advice on my future steps and decisions. Moreover, Kevin has provided me deep and comprehensive understanding in many deep learning topics.

I would also like to thank Noel O'Connor and the rest of the team at Insight Center for Data Analytics for allowing me to do this project and providing very useful advice.

Last but not least, I want to thank my family, friends, and Clara for the strong support I've always recieved.

Revision history and approval record

Revision	Date	Purpose
0	12/06/2017	Document creation
1	23/06/2017	Document revision
2	26/06/2017	Document revision
3	29/06/2017	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Marc Assens Reina	marc.a.r95@gmail.com
Xavier Giró i Nieto	xavier.giro@upc.edu
Kevin McGuinness	kevin.mcguinness@insight-centre.org

Written by:		Reviewed and approved by:		Reviewed and approved by:	
Date	23/06/2017	Date	29/06/2017	Date	29/06/2017
Name	Marc Assens Reina	Name	Xavier Giró i Nieto	Name	Kevin McGuinness
Position	Project Author	Position	Project Supervisor	Position	Project Supervisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Teaching computers where to look

One of the most important features of the human species is the ability to process and understand visual information from the environment. Computer Vision (CV) is the research field that studies how computers can be made for gaining high-level understanding from digital images or videos. It seeks to mimic tasks that the human visual system can do such as object recognition, video tracking, semantic segmentation, saliency prediction, etc. This project focuses on the subdomain of saliency prediction (sometimes called visual attention).

The field of saliency estimation consists of predicting human eye fixations, and highlights regions of interest for human observers. The positions where human observers look at an image provides insight about human scene understanding: what are the important parts, where actions are happening, what participants are involved, etc. Saliency estimation is a relevant field because it provides insight into the human visual system, and improves other CV related tasks such as object recognition [1]. A recent study suggests that when people are captured on an image, human observers spend time trying to figure out where are those people looking and why [4].

If you think of humans as if they are computers, when a human observes an image, he shifts his gaze and spends more time observing certain subsets of the visual input. Therefore, allocating more processing resources to those subsets. Saliency prediction studies how humans allocate processing resources for visual input [5].

There are two main ways of representing saliency information: *saliency maps* and *scanpaths*. Saliency maps represent the probability of each corresponding pixel in the image of capturing human attention and have received much attention by the research community over the last years. If saliency maps tell *where* do humans look, scanpaths tell *when* do humans look at each of the relevant regions of the visual input. They represent a sequence of timestamped and ordered fixations on an image. There has been little research on this later topic.

Many researchers stress the urgent need to investigate scanpaths [37][8][17][16][36]. Moreover, recent advances in fields like Virtual Reality and Augmented Reality have arisen the need for models estimate scanpaths for rendering purposes.

If we want computers to be able to see like humans, we have to teach them to allocate processing resources on the right places, at the right time.

This thesis has focused on adding the temporal dimension to saliency prediction. The main contributions of this thesis are:

- **Insight about scanpath data** We have found that individual scanpaths have a very stochastic nature and their beginning is usually biased towards the center.
- **A model for scanpath prediction** We propose a model that is able to generate scanpaths from any given image.

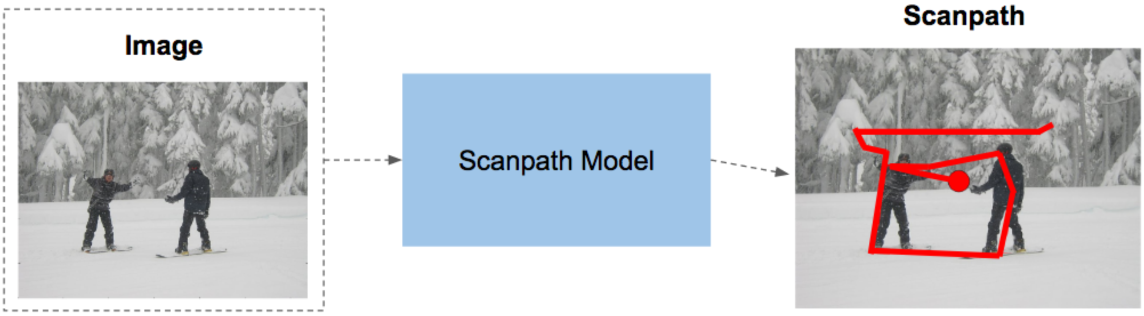


Figure 1.1: High-level architecture of the model that we wanted to build. The example shown is a real sample of the SALICON dataset.

- **Saliency Volumes** A novel time-aware saliency representation named *Saliency Volume*, that averages scanpaths over many observers. Scanpaths can be extracted from this representation.
- **Implementation of SalNet with the Keras framework** In the process of creating our model, we found the need to port the architecture of SalNet[30] (originally implemented in Caffe) to Keras and retrain it from scratch. This is a popular saliency prediction model, and the Keras implementation had already been asked by some members. The code is live in a github repository[1].
- **Contribution to an open source project** We implemented the Meanshift algorithm using a multivariate Gaussian kernel in a popular open source repository[2].

1.2 Requirements and Specifications

The requirements for this project are:

- **Understand the data** Explore and analize the data of the available datasets and gain insights to develop better models.
- **Create a model** Develop software that can generate scanpaths from any given image.
- **Participate in the Salient360! Challenge** Adapt our model to predict scanpaths for 360 images and present it to the Salient360! Challenge that takes place at the ICME conference

[1] todo

[2] https://github.com/mattnedrich/MeanShift_py/pull/3

- If the results are satisfactory, contribute to the dissemination of this work.

The specifications of this project are the following:

- Use the programming language *Python* in the best possible way.
- Use the deep learning frameworks *Keras* and *Tensorflow* to implement the models.
- Create a replicable programming environment using *Docker*. This allows much better portability and reusability of the project and code.

1.3 Work Plan

This project was developed in collaboration with the GPI research group[3] of the Universitat Politècnica de Catalunya and the Insight Center for Data Analytics[4] of the Dublin City University. Weekly meetings were held with the supervisors to disscuss the progress and decisions made.

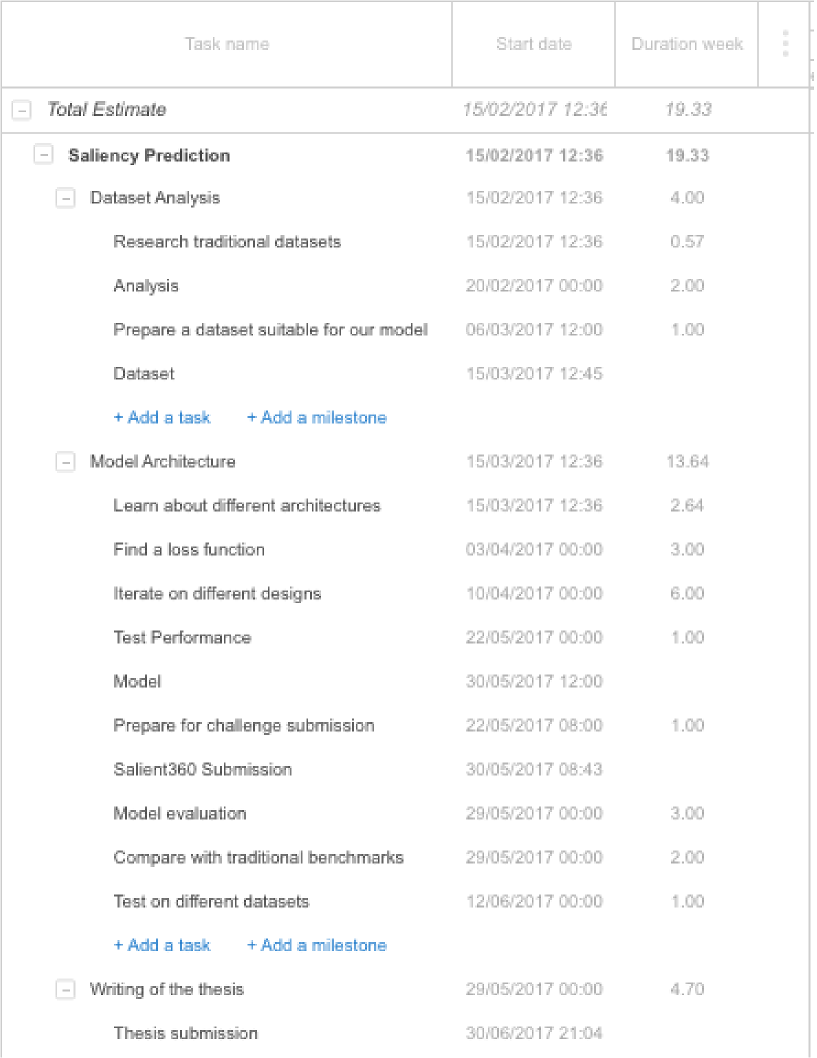
Below we present the work plan and its deviations from the original plan. The changes made to the work plan are detailed in section *1.4 Incidents and Modifications*.

1.3.1 Work Packages

- WP 1: Dataset analysis
- WP 2: Model architecture
- WP 3: Performance test
- WP 4: Writing of the thesis

[3] https://imatge.upc.edu/web/

[4] https://www.insight-centre.org



Task name	Start date	Duration week
Total Estimate	*15/02/2017 12:36*	*19.33*
Saliency Prediction	**15/02/2017 12:36**	**19.33**
Dataset Analysis	15/02/2017 12:36	4.00
Research traditional datasets	15/02/2017 12:36	0.57
Analysis	20/02/2017 00:00	2.00
Prepare a dataset suitable for our model	06/03/2017 12:00	1.00
Dataset	15/03/2017 12:45	
+ Add a task + Add a milestone		
Model Architecture	15/03/2017 12:36	13.64
Learn about different architectures	15/03/2017 12:36	2.64
Find a loss function	03/04/2017 00:00	3.00
Iterate on different designs	10/04/2017 00:00	6.00
Test Performance	22/05/2017 00:00	1.00
Model	30/05/2017 12:00	
Prepare for challenge submission	22/05/2017 08:00	1.00
Salient360 Submission	30/05/2017 08:43	
Model evaluation	29/05/2017 00:00	3.00
Compare with traditional benchmarks	29/05/2017 00:00	2.00
Test on different datasets	12/06/2017 00:00	1.00
+ Add a task + Add a milestone		
Writing of the thesis	29/05/2017 00:00	4.70
Thesis submission	30/06/2017 21:04	

Figure 1.2: Work packages

1.3.2 Gantt Diagram

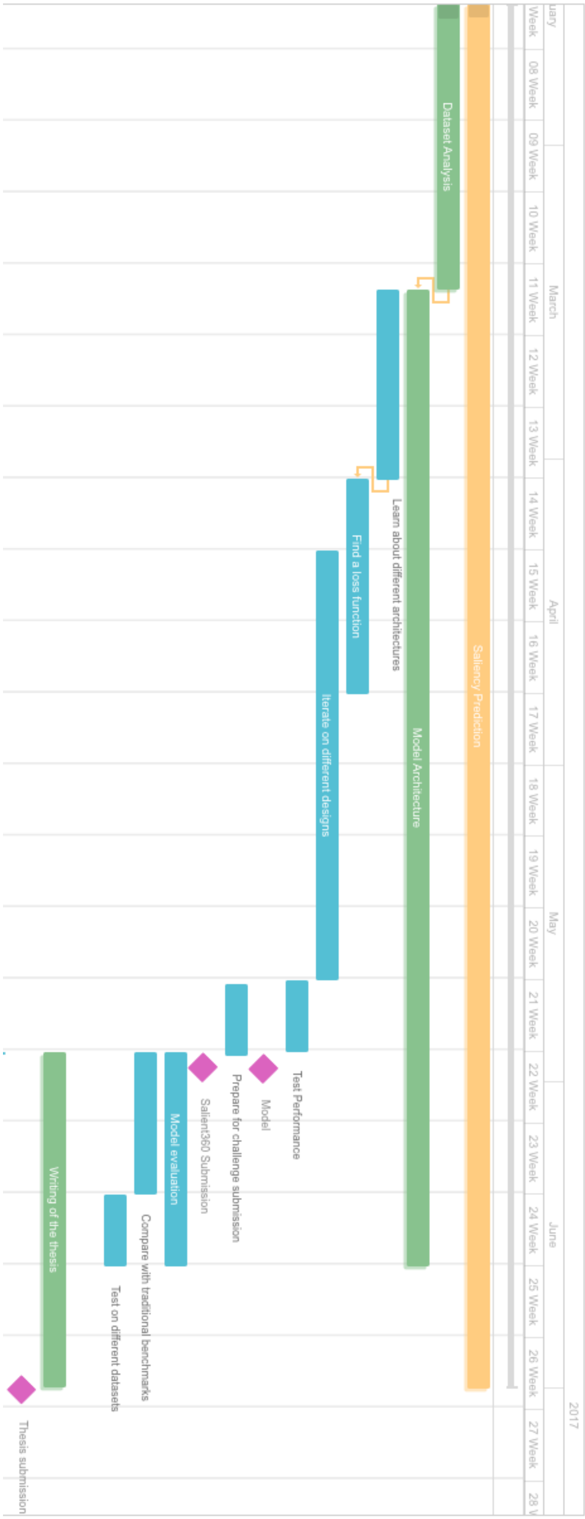


Figure 1.3: Gantt Diagram of the Degree Thesis

1.4 Incidents and Modifications

The work plan has suffered some changes as the project has advanced.

- **Apply to the Salient360! Challenge** We decided to apply to one of the Grand Challenges of the IEEE International Conference on Multimedia and Expo 2017 (ICME). The submission of the model was added to the work plan as a milestone.
- **Added a work package for writing** Although all the progress was documented on a weekly basis, I decided to create a separate work package to allocate time for the writing of the thesis. This work package included sending preliminary versions of the thesis to different persons for review.

Chapter 2

State of the art

Predicting where a human observer might fixate when viewing an image in a freeviewing scenario has been of interest in the computer vision community for the last decades. Significant progress has been made due to simultaneous advances in computing infrastructure, data gathering, and algorithms.

2.1 Saliency prediction

The first predictive models were biologically inspired and based on a bottom-up computational model that extracted low-level visual features such as intensity, color, orientation, texture and motion at multiple scales. Itti et al. [13] proposed a model that combines multiscale low-level features to create a saliency map. Harel et al. [9] presented a graph-based alternative that starts from low-level feature maps and creates Markov chains over various image maps, treating the equilibrium distribution over map locations as activation and saliency values.

Although these models did well qualitatively, they had limited use because they frequently did not match actual human saccades from eye-tracking data. It seemed that humans not only base their attention on low-level features, but also on high-level semantics [4] (e.g., faces, people, cars, etc.). Judd et al. introduced in [18] an approach that used low, mid and high-level image features to define salient locations. This features were used in combination with a linear support vector machine to train a saliency model. Borji [2] also combined low-level features with top-down cognitive visual features and learned a direct mapping to eye fixations using Regression, SVM and AdaBoost classifiers.

Recently, the field of saliency prediction has made significant progress due to the advance of deep learning, and its applications on the task of Image Recognition [19] [33]. This advances suggest that these models can capture high-level features. As stated in [4], in March of 2016 there were six deep learning models among the top 10 results in the MIT300 saliency Benchmark [3]. In [25] Pan et al. compared shallow and deeper CNNs. Following the idea of modeling bottom-up and top-down features at the same time, Liu et al. [26] presented a multiresolution convolutional neural network (Mr-CNN) that combines predictions at different resolutions using a final logistic regression layer to predict a saliency map.

The enormous amount of data necessary to train these networks makes them difficult to learn directly for saliency prediction. With the aim of allowing saliency models to capture these high-level features, some authors have adapted well-known models with good performance in the task of Image Recognition (this technique is called *transfer learning*). DeepGaze [20] achieved state of the art performance by reusing the well-known AlexNet [19] pretrained on ImageNet [7] with a network on top that reads activations from the different layers of AlexNet. The output of the network is then blurred, center biased and converted to a probability distribution using a softmax. A second version called DeepGaze 2 [22] used features from VGG-19 [34] trained for image recognition. In this case, they did not fine-tune the network. Rather, some readout layers were trained on top of the VGG features to predict saliency maps with the SALICON dataset [15]. These results corroborated that deep architectures trained on object recognition provide a versatile feature space for performing related visual tasks.

In [35], Torralba et al. studied how different scenes change visual attention and discovered that the same objects receive different attention depending on the scene where they appear (i.e. pedestrians are the most salient object in only 10% of the outdoor scene images, being less salient than many other objects. Tables and chairs are among the most salient objects in indoor scenes). With this insight, Liu et al. proposed DSCLRCN [25], a model based on CNNs that also incorporates global context and scene context using RNNs. Their experiments have obtained outstanding results in the MIT Saliency Benchmark.

Recently, there has been interest in finding appropriate loss functions. Huang et al. [11] made an interesting contribution by introducing loss functions based on metrics that are differentiable, such as NSS, CC, SIM and KL divergence to train a network (see [32] and [21]).

Other advances in deep learning such as generative adversarial training (GANs) and attentive mechanisms have also been applied to saliency prediction: Pan et al. recently introduced SalGAN [29], a deep network for saliency prediction trained with adversarial examples. As all other Generative Adversarial Networks, it is composed of two modules, a generator and a discriminator, which combine efforts to produce saliency maps. Cornia et al. presented in [6] a model that incorporates neural attentive mechanisms. The model includes a Convolutional LSTM that focuses on the most salient regions of the image to iteratively refine the predicted saliency map. Additionally, they tackle the center bias present in human eye fixations by learning a set of prior map produced by Gaussian functions.

2.2 Scanpath prediction

In contrast with the related task of saliency map prediction, there has not been much progress in the task of scanpath prediciton over the last years. Cerf et al. [5] discovered that observers, even when not instructed to look for anything particular, fixate on a human face with a probability of over 80% within their first two fixations. Furthermore, they exhibit more similar scanpaths when faces are present. Recently, Hu et al. [10] have introduced a model capable of selecting relevant areas of a 360 video and deciding in which direction should a human observer look at each frame. An object detector is used to propose candidate objects of interest and a RNN selects the main object at each frame.

We believe this line of research is meaningful and it has not received much attention by the research community over the last years.

2.3 Metrics

As stated in section *2.1 Saliency Prediction*, most of the research has focused on generating and measuring *saliency maps* (also known as *attention maps*). This representations can be evaluated using a broad range of metrics, which all ignore sequence and temproral information.

In contrast, metrics that take into account sequence and temporal information have received less attention, and we can differentiate three types:

- **String-edit measures** It replaces fixations that are in areas of interest (AOIs) with characters that form a string that can be compared. The similarity between two or more strings is reduced to counting edit operations (insertions, deletions, or substitutions). This measure only takes into account the order of the fixations, but not the time. In this category, the Levenshtein distance [24] is the most common.
- **Mannan distance** It compares scanpaths by their spatial properties rather than time, and the order of fixations is completely ignored. It measures the similarity between scanpaths by calculating the distance between each fixation in one scanpath and its nearest neighbor in the other scanpath [27].
- **Vector based** Jarodzka et al. [14] has recently proposed a metric that views scanpaths as a sequence of geometric vectors that correspond to the saccades of the scanpath. This similarity metric compares scanpaths across several dimensions: shape, fixation position, length, direction, and fixation duration.

In this project, we have evaluated our models using the vector-based metric proposed by Jarodzka. This is also the metric used in the 360 Salient Challenge.

Chapter 3

Methodology

This section presents the methodology used to develop this project and explains how we produced the results. It also provides insight into the decisions that were made to design the final architecture of the model.

3.1 Setting up the working environment

3.1.1 Programming languages and frameworks

The main languages we initially considered were Python, Lua, C++, and Octave. Nevertheless, several frameworks written in Python like Theano or Tensorflow have appeared over the last few years, and it seems the research and developer community is moving towards them. We decided to use the Python programming language to code the project. We considered this was the best programming language regarding documentation available and development speed.

We were also recommended to use the high-level framework Keras to speed up prototyping. Keras provides implementations and examples of the leading deep learning architectures, and it can be used with Tensorflow or Theano as backend. In some cases, we also had to use pure Tensorflow to accomplish the tasks.

3.1.2 A reproducible environment with Docker

Setting up a working environment for deep learning research is challenging because the frameworks are constantly changing, and all the computation has to be processed by a GPU. We used Docker[1] to isolate our working environment and make it reproducible on any computer.

3.1.3 A command line utility for fast prototyping

After one month of research, the number of carried out experiments grew, and we faced some challenges regarding the organization of our experiments and their respective inputs and outputs. We decided to create a command line utility that helped organize our prototyping workflow. We called it *TidyBot*, and it is available at `https://github.com/massens/tidybot`.

The command `$ tidybot init` creates a folder structure for each experiment containing */input*, */output*, */testing*, */evaluation*. The command `$ tidybot model` creates a new Python file with some boilerplate code, and creates I/O folders for this model under */input* and */output*.

[1] https://www.docker.com

The folder */testing* was included taking inspiration from Test Driven Development [2] (TDD). For each model, a Jupyter Notebook is created inside that folder to test that all the parts are working correctly before the model is trained.

3.2 Understanding the data

To the knowledge of the authors, the task of scanpath prediction has not been addressed before. Therefore, we decided that the first step towards building a model was to analize the properties of scanpaths and understand what data is available. We chose datasets that have been previously used for the related task of saliency map prediction such as iSUN [39] and SALICON [15].

3.2.1 Properties of scanpaths

After analysing the datasets using Jupyter Notebooks, we observed the following properties:

- **Scanpaths have a stochastic nature** If we explore individual scanpaths qualitatively, we observe that they do not always fixate in very salient regions.
- **Scanpaths of different users can be very different** If we compare scanpaths from various users produced with the same stimuli we observe that while they can share a similar pattern, they can also be very different.

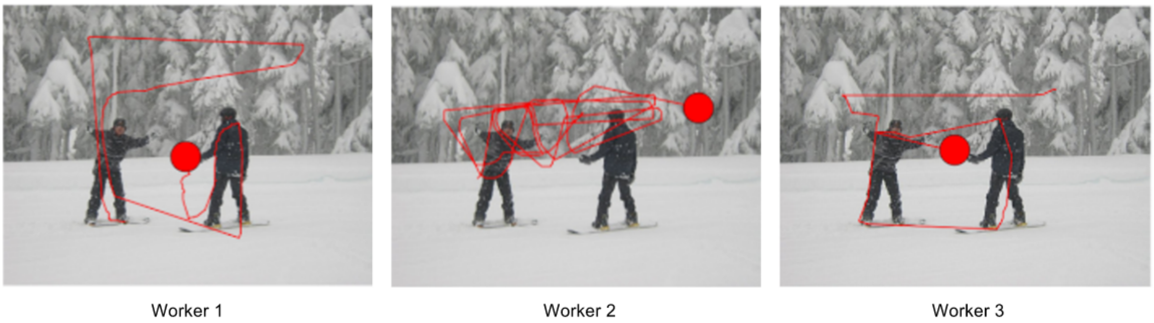


Figure 3.1: Scanpaths of different users on an example image of the SALICON dataset

- **Center bias of the first fixation** Most of the users start their fixations at the center of the picture. This property has also been observed by other authors [6].

[2] https://www.agilealliance.org/glossary/tdd/

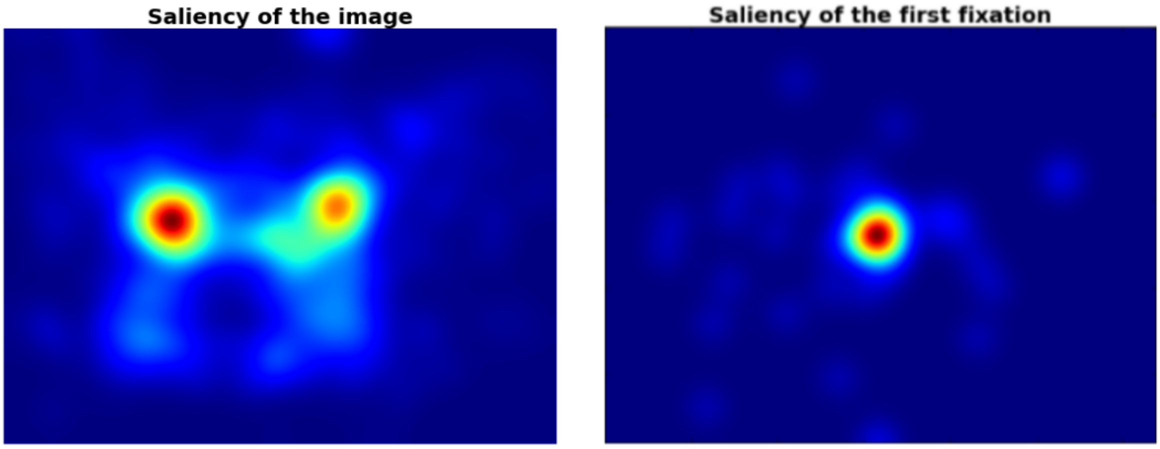


Figure 3.2: Architecture of Object-Scene Convolutional Neural Network for event recognition

- **Scanpaths diverge and converge over time** We observed that while most users start at the same place, their fixations can diverge and converge while they explore the image.

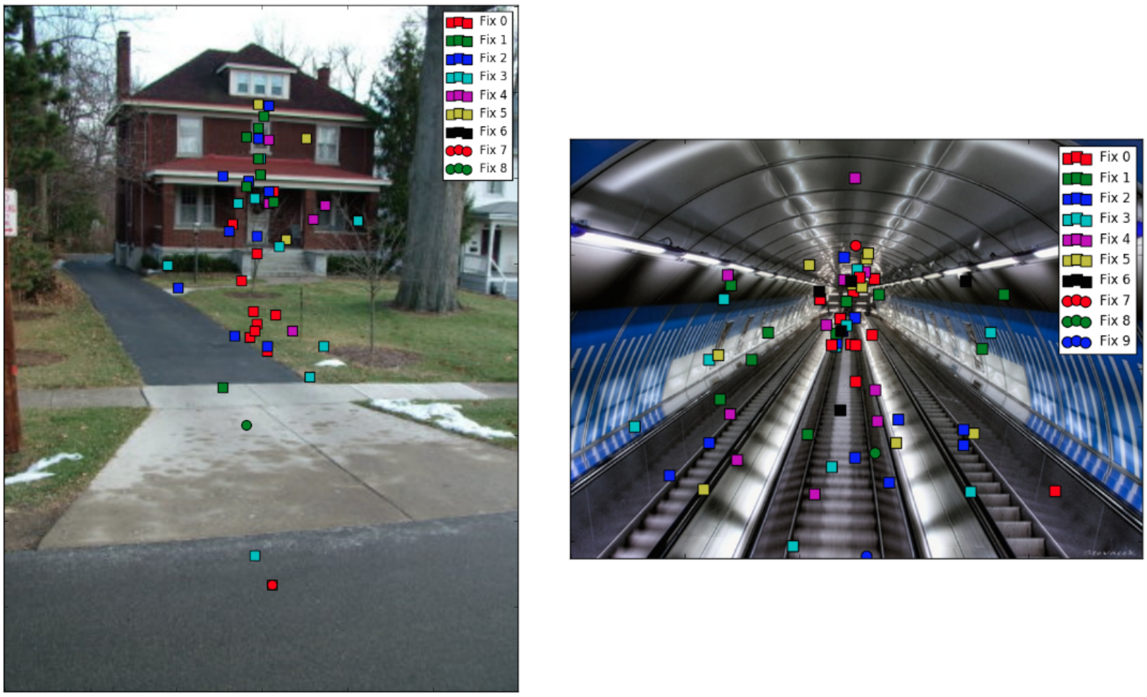


Figure 3.3: Above we have fixation maps, where colors represent the number of each fixation (i.e. red squares represent the first fixation of different users). The image on the left shows that users fixate first on the center, and the second fixation is usually at the top of the image. Finally, different users start looking at different places. On the picture on the right, we observe that while most of the users fixate on the center at the beginning, the 2nd-4th fixations are very different across users. Nevertheless, at the end of the scanpaths they tend to return to the center (red circles and black squares).

- **We can average multiple scanpaths to get a consistent representation** Given that scanpaths have a high random component, we can average them to obtain a more consistent representation. This concept is explained in detail in section *3.4.1 Introducting saliency* *volumes.*

- **How many scanpaths do we need for a consistent representation?** It is very imperative to know how many scanpaths are needed to produce a consistent representation. Our results show that we need as little as 3-4 scanpaths. More information in the section *3.4.1 Introducing saliency volumes.*

3.2.2 Differences between saliency maps and scanpaths

A saliency map is a single-channel image that represents the probability of each point being fixated by a user. It is generated by aggregating fixation points of different users and convolving with a Gaussian kernel. Therefore, they present saliency information averaged from multiple users. The MIT benchmark of saliency [3] predicted how well a single user predicts fixations of a group of users. As depicted in Figure 3.4, they found that there is a gap between the accuracy of the prediction of a single user, and the accuracy of a prediction averaged over multiple users.

This suggests that in the domain of saliency prediction, it might make more sense to work with representations that average fixations over multiple users.

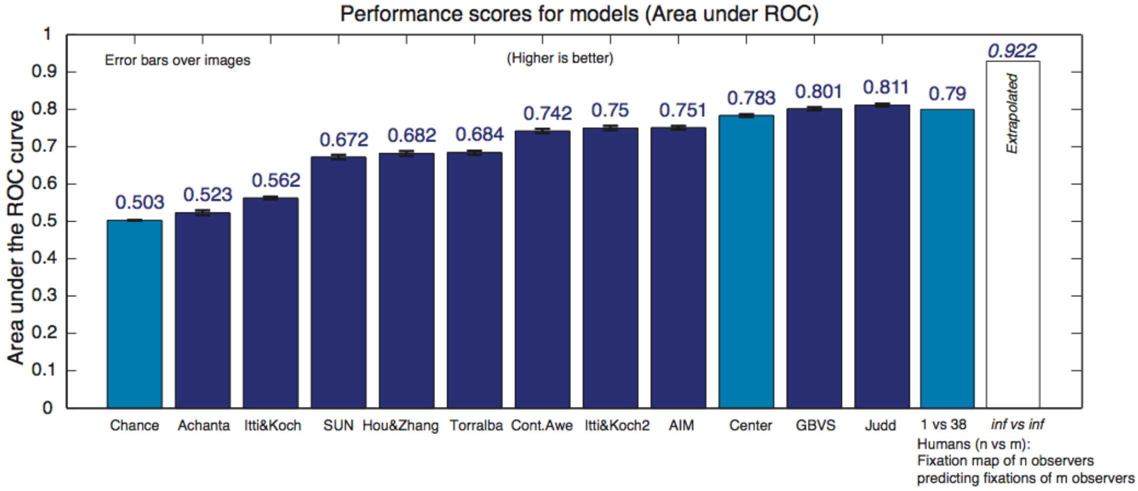


Figure 3.4: Accuracy of different models measured with the ROC metric. Source: MIT Benchmark of Saliency.

In contrast, scanpaths represent the sequence of saccades that a single user generates for a given stimuli. It is not averaged over multiple users.

3.3 A toy model to directly predict scanpaths

With the knowledge we had gained in the section *3.2 Understanding the Data*, we decided to implement a supervised deep learning model that predicts scanpaths using an off-the-shelf network and LSTMs.

3.3.1 Dataset and preprocessing

To train this model we used the 6, 000 images from the iSUN Dataset [39]. This dataset provides all the raw eye positions recorded as well as fixation points (clusters of positions) obtained with the Meanshift algorithm[3]. While the positions are timestamped, the fixation points are provided without order (they have an intrinsic order, but it was not included in the dataset by the authors).

It was important to retrieve the order of the fixation points because our model had to learn from a sequence of ordered fixation points. We used the same method used by the authors to associate each fixation point with its respective timestamped eye positions. Then, we calculated the timestamp of each fixation point by averaging the timestamps of its respective positions. Finally, we ordered the fixation points with the calculated timestamps.

The authors used the Meanshift algorithm with a multivariate Gaussian kernel, but we could not find an implementation online with this characteristics. We had to implement it ourselves, and the code in Python was committed to a popular open source repository [4].

Moreover, the pixel values of the images were centered by substracting to each channel the mean of the dataset's channel. The fixation points were normalized to [0,1] to be able to use a sigmoid activation. Some experiments were done normalizing the fixation points to [-1, 1] and using a linear activation at the end of the network, but they did not show a performance increase.

3.3.2 Model architecture

This network consists of three main blocks: 1) extracts features with VGG16, 2) Convolutional LSTMS and 3) a fully-connected layer with two units and linear activation. The two units of the last layer represent the two components of a spatial coordinate (x,y). The scanpath problem is treated as a regression problem and the Mean Squared Error loss function is used for training.

This network has a total of 72.3 million free parameters. The optimizer used was stochastic gradient descent with a learning rate of 0.001. The input images are resized to $[224 \times 224]$ to decrease the number of free parameters. We used a batch size of 2 images to avoid overflowing the memory of our GPU.

[3] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html

[4] `https://github.com/mattnedrich/MeanShift_py/pull/3`

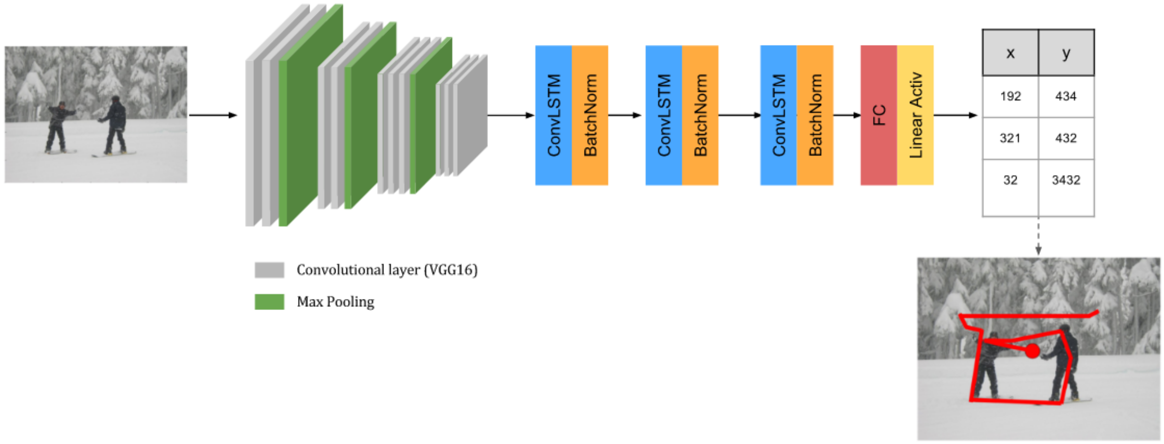


Figure 3.5: First model architectre

3.3.2.1 Transfer learning with VGG16

When there is not a lot of training data available, it is a common practice to reuse a pre-trained model for a similar task. In this case, we decided to use the well known VGG16 [34] network trained for image recognition. This model uses 16 weighted layers and has 138 million parameters. This allows us to reuse low level and high-level features. We used the keras implementation of VGG16[5].

During training, we fine-tuned this layers with our parameters.

3.3.2.2 Convolutional LSTMs

The feature maps extracted using the VGG16 network are fed into a sequence of three stacked Convolutional LSTMs [38] (ConvLSTMs). These layers are responsible for generating a sequence of fixation points that form a scanpath. They have 'memory,' and each fixation points is conditioned to all the previous ones. Moreover, they are capable of generating a variable number of fixation points.

We decided to use Convolutional LSTMs instead of regular LSTMs do decrease the number of parameters of the model. Using a regular LSTM increased the number of parameters to the point that the model did not fit into memory. It is a common practice to alternate ConvLSTMs with batch normalization[12] layers to accelerate the training.

3.3.3 Where to go from here?

The observation of the outputs of this first toy example gave us very interesting insight. We observed that the model usually predicts most of the fixation points in the center of the image. This suggests that due to the stochastic nature of scanpaths (mentioned in *3.2.1 Properties of Scanpaths*), there are many possible scanpaths a single image. The use of the MSE loss results

[5] https://github.com/fchollet/keras/blob/master/keras/applications/vgg16.py

in predicting an average scanpath over possible scanpaths, instead of a possible scanpath. This effect has also been recently studied by Yann LeCun et al. [28].

Nevertheless, we want a possible scanpath. Not an averaged and non-realistic scanpath.

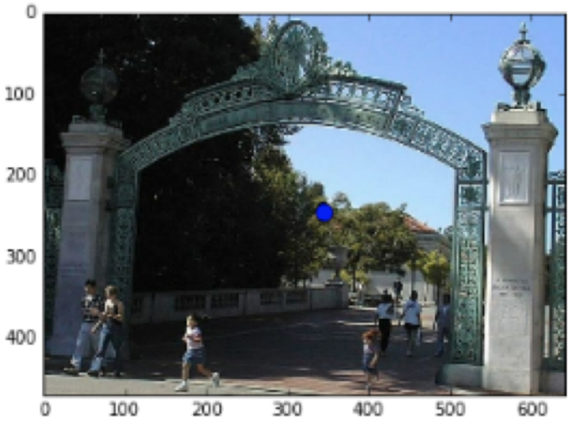


Figure 3.6: The model learns very fast to predict most of the fixation points of the scanpath in the center of the image.

With this insight, we evaluated two new possibilities:

1. **Adversarial Training** is an approach that has already proved useful in similar situations [28]. It is easy for an adversary to detect unrealistic examples. The downside of this method is that adversarial models are usually difficult to train and it might not have been a wise decision given our time constraints.
2. **Predicting a less stochastic representation** There are reasons to search a time-aware saliency representation that describes the mean behavior of multiple users. In other words, that describes *where* in the image, and *when* most of the users look. This representation might be more useful than scanpaths.

Our final decision was to find a less stochastic representation and will be presented in section *3.4.1 Introducing saliency volumes*. A model should be more successful at predicting this representation, as it is more consistent than single user scanpaths. We decided to predict a volume instead of using an adversarial model due to our tight schedule, and because we believe it can be useful in many applications. Besides, we will have to find the best way of generating scanpaths from this new saliency representation.

3.4 A model that predicts saliency volumes

With the insight gained from the previous model, we propose a novel time-aware saliency representation named *Saliency Volume* that is capable of expressing saliency information with spatial and temporal dimensions. In other words, it can tell *where* and *when* do users look at an image. Then, we propose an architecture based on SalNet [30] that is capable of predicting saliency volumes. Finally, we present different strategies to generate individual scanpaths from saliency volumes.

This model was trained on the SALICON dataset, which has 9000 training examples. We used this dataset because its data facilitated the generation of saliency volumes.

3.4.1 Introducing: Saliency Volumes

Saliency volumes aim to be a suitable representation of spatial and temporal saliency information for images. They have three axes that represent the width and height of the image, and the temporal dimension.

Saliency volumes are generated from information already available in current fixation datasets. In these datasets, each fixation has a position (width, height) and a timestamp. First, the timestamps of the fixations are quantized. The length of the time axis is determined by the longest timestamp and the quantization step. Second, a binary volume is created by placing '1' on the fixation points and '0' on the rest of the positions. Third, a multivariate Gaussian kernel is convolved with the volume to generate the saliency volume. The values of each temporal slice can be normalized, converting the slice into a probability map that represents the probability of each pixel being fixated by a user at each timestep.

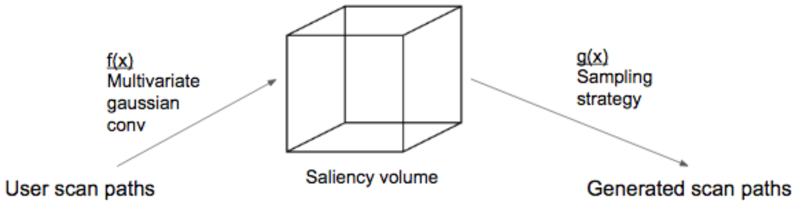


Figure 3.7: Saliency volumes can be generated from fixations, and scanpaths can be generated from saliency volumes.

Figure 3.8 shows how saliency volumes are a meta-representation of saliency information and other saliency representations can be extracted from them. Saliency maps can be generated by performing an addition operation across all the temporal slices of the volume and normalizing the values to ensure they add to one. A similar representation is *temporally weighted saliency maps*, which are generated by performing a weighted addition operation of all the temporal slices. Finally, scanpaths can also be extracted by sampling fixation points from the temporal slices. Sampling strategies that aim to generate realistic scanpaths are will be discussed in the *3.4.3 Sampling strategies to generate scanpaths from saliency volumes*.

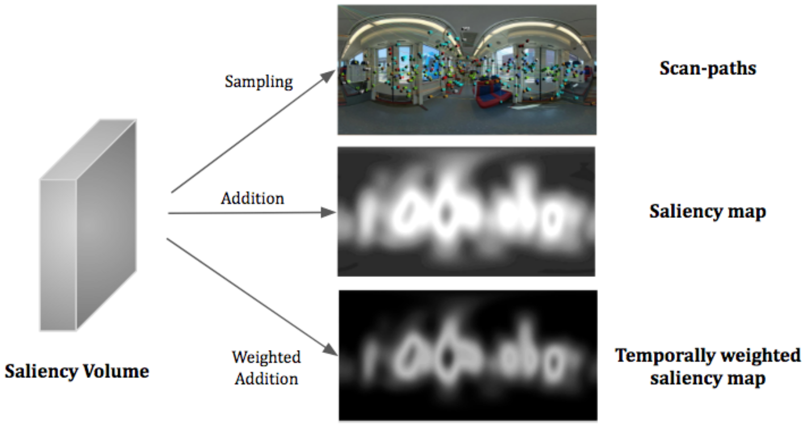


Figure 3.8: Scanpaths, saliency maps, and temporally weighted saliency maps can be generated from a saliency volume.

3.4.2 Model architecture

Once we have generated the saliency volumes from the data available in the iSUN dataset, they are represented as matrices with three dimensions. These matrices are the target of our model. Our model is based on an encoder-decoder architecture.

This network adapts the filters learned to predict flat saliency maps to predict saliency volumes. Figure 3.9 illustrates the architecture of the convolutional neural network, composed of 10 layers and a total of 25.8 million parameters. Each convolutional layer is followed by a rectified linear unit non-linearity (ReLU). Excluding the last three layers, the architecture follows the proposal of SalNet [30], whose first three layers were at the same time extracted from the VGG-16 model [34] trained for image classification. Our final sigmoid layer has three dimensions corresponding to the ones of the saliency volume.

Originally, SalNet used the deep learning framework Caffe, which nowadays lacks many features of modern DL libraries. Because of this, we decided to port the architecture of SalNet to Keras and retrain it. The code and weights can be found at `https://github.com/massens/salnet-keras`[6].

The model was designed considering the amount of training data available. Different strategies were introduced to prevent overfitting. The model was previously trained on the similar task of saliency map prediction, and the obtained weights were fine-tuned for the task of saliency volume prediction. We used the stochastic gradient descent optimizer with a learning rate of 0.001, and a batch size of two samples.

[6] This repository also has intrinsic value, as there have already been requests for a Keras implementation of SalNet by the community.

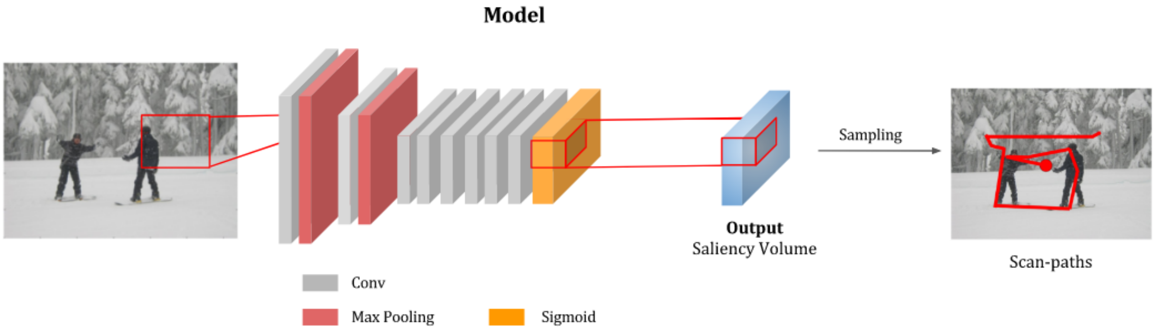


Figure 3.9: Architecture of our second model.

3.4.3 Sampling strategies to generate scanpaths from saliency volumes

To sample scanpaths from a saliency volume we have to take into consideration a couple of things. Firstly, we have to decide how many fixations will a scanpath have. Then, each fixation has to have a position (width, height), and a duration (in seconds). To decide the number of fixations and their durations, we sampled values from the data distribution on the training set plotted in Figure 3.10.

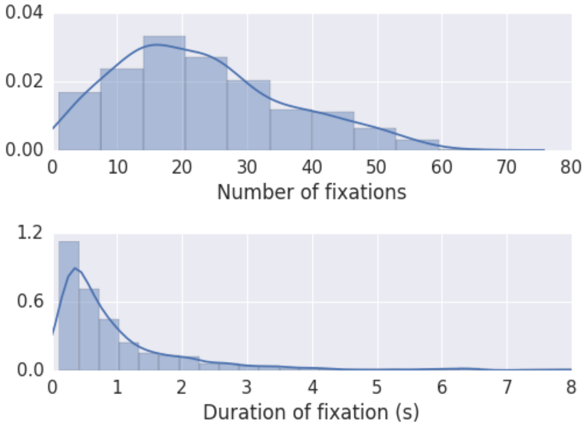


Figure 3.10: Probability distribution of the number of fixations per scanpaths (top) and duration of each fixation (bottom).

Regarding the spatial location of the fixation points, three different strategies were explored:

1. **Naive sampling strategy** The simplest approach consists of taking one fixation for each temporal slice of the saliency volume. Through qualitative observation, we noticed that scanpaths were unrealistic, as the probability of each fixation is not conditioned to previous fixations.

2. **Limiting distance between fixations** When we look at images, each fixation we generate is usually close to previous one. Thus, a more elaborated sampling strategy consists of forcing fixations to be closer to their respective previous fixation. This is accomplished by multiplying a temporal slice (probability map) of the saliency volume with a Gaussian

kernel centered at the previous fixation point. This suppresses the probability of positions that are far from the previous fixation point.

3. **Avoiding fixating on same places** It is reasonable to think that if we have already fixated on an area, we won't fixate again. The third sampling strategy we assessed consisted on suppressing the area around all the previous fixations using Gaussian kernels.

As we will discuss in section *4.4.1 Creating a baseline*, our results show that the best performing sampling strategy is the second one: limiting distance between fixations.

3.5 Final model: fine tuning for 360 images

We decided to fine tune our model with the purpose of participating at the *Salient360!: Visual attention modeling for 360 Images Grand Challenge* [7] from the IEEE International Conference on Multimedia and Expo 2017.

This meant fine tuning our second model (section *3.4 A model that predicts saliency volumes*) with the dataset provided by the Salient360 organization, and adding little improvements to the architecture. The different blocks are depicted in Figure 3.11.

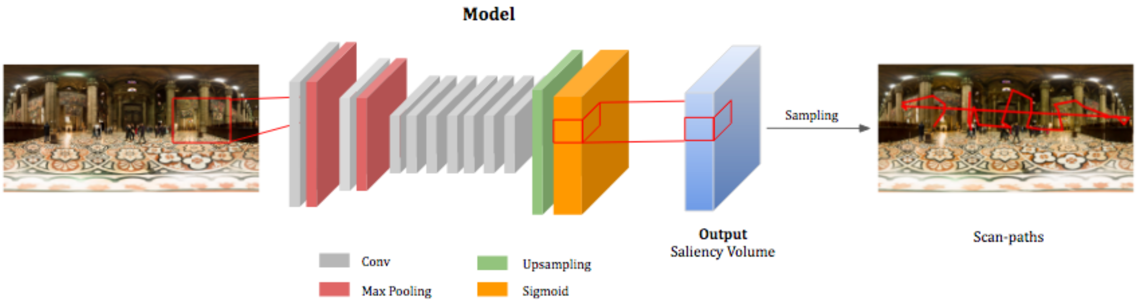


Figure 3.11: Architecture of our third model that is able to predict scanpaths from omni-directional images.

3.5.1 Dataset

To train a model using supervised learning we need a tremendous amount of data. This data are examples of input-output pairs. In this case, the inputs are omni-directional images (360 images), and the outputs are saliency volumes.

For this last model, we have used the dataset provided by the Salient360 organization, which is the first dataset of omni-directional images for saliency prediction. Each data sample is an input-output pair, where the input is an omni-directional image, and the output is a group of forty scanpaths. The training subset is composed of 40 images, along with heat maps and 1500 scanpaths; the test set is composed of 25 images and 1000 scanpaths. The whole explanation on how the dataset was created and how it is organized can be found in their paper [31].

[7] http://www.icme2017.org/grand-challenges/

The fact that this dataset is very small makes it only suitable for fine tuning a pre-trained model, and not for training a model from scratch.

3.5.2 Transfer learning

Because the dataset only has 40 training examples, we had to perform transfer learning. We reused the model trained in section *3.4* and added a few more layers at the end.

During training, the weights of the first layers were initialized with the ones obtained in section 3.4, while the weights of the last two layers were initialized randomly. Moreover, due to the lack of training samples, we did not fine tune the parameters of the first layers to avoid overfitting the training set.

3.5.3 Deconvolutional layers

The original architecture of SalNet [30] used a learned upsample (deconvolution layer) to produce an output with the same dimensions as the input. This seemed to be a reasonable feature to add to our model. As depicted in Figure 3.11, the deconvolutional layer is nothing more than an upsampling layer followed by a convolutional layer. In this case, the upsampling layer had a stride of 4, and the convolutional layer had a kernel size of 9.

Secondly, the input images were resized to $[300 \times 600]$, a much smaller dimension than their initial size $[3000 \times 6000]$. The last layer of the network outputs a volume with size $[12 \times 300 \times 600]$, with three axis that represent time, and height and width of the image.

Chapter 4

Results

The models presented in *3. Methdologies* for scanpath prediction were assessed and compared from different perspectives. First, we evaluate the impact of different sampling strategies to generate scanpaths from saliency volumes. Finally, we show quantitative and qualitative performance results of the model.

4.1 Evaluation Metric

Scanpath prediction evaluation has received attention lately and it is a very active field of research [23][14]. For this work, we have used a metric proposed by the Salient 360 Challenge organization that compares the similarity of 40 generated scanpaths with the ground truth scanpaths. The similarity metric used is the Jarodzka algorithm, where the similarity criteria was slightly modified to use equirectangular distances in 360 degrees instead of euclidean distances. Also, the generated and ground truth scanpaths are matched 1 to 1 using the Hungarian optimizer to get the least possible final cost.

The Jarodzka algorithm views scanpaths as a sequence of geometric vectors that correspond to the saccades of the scanpath. This similarity metric not only compares scanpaths on the spatial dimension, but also on any dimension available in saccade vectors (shape, fixation position, length, direction, and fixation duration).

The algorithm has two parts: 1) temporal alignment of scanpaths, and 2) Scanpath comparison.

4.1.1 Temporal alignment of scanpaths

In this first step, scanpaths are temporally aligned to each other based on their shape and characteristics. We will denote the two scanpaths as a series of vectors (corresponding to the saccades) $S_1 = \{v_1, v_2, ..., v_n\}$ and $S_2 = \{u_1, u_2, ..., u_n\}$.

The alignment is performed following these steps:

1. **Saccade similarity matrix** Compute how similar each saccade is to the others using a similarity metric such as saccade amplitude. The results are collected in a matrix $M(i, j)$.
2. **Create a graph** where the matrix elements are called nodes.
3. **Find the shortest path** from the node 1 to the last node in the graph using the Dijkstra algorithm [Dijkstra 1959][1]. Align scanpaths along the shortest path. On the original paper

[1] https://www.ssucet.org/old/pluginfile.php/2121/mod_resource/content/1/21-dijkstra.pdf

the scanpaths are aligned based on their shapes. Nevertheless, the alignment may be performed in other dimensions.

4.1.2 Scanpath Comparison

For each pair of fixation and saccade vectors the following measures are computed (average):

1. **Difference in shape** Computed using $u_i - v_j$
2. **Difference in amplitude** between saccade vectors $||u_i - v_j||$
3. **Distance between fixations**
4. **Difference in direction (angle)** between saccade vectors
5. **Difference in duration between fixations** between saccade vectors

These measures indicate how similar two scanpaths are along different dimensions. With the objective of obtaining a unique similarity metric with values between $[0, 1]$, the first three measures are normalized by the screen diagonal. Direction is normalized by π, whereas the difference in duration is normalized against the maximum duration of the two durations compared

4.2 Model architectures and setups

In the following sections, we will refer to the different models developed by an identifier. The descriptions of the models and their respective identifiers are found in Table 4.1.

id	Description of the model
0	The first toy model that aims to directly predict scanpaths. Described in 3.3 and schema in Fig 3.5.
1	Model based on SalNet that predicts saliency volumes. Described in 3.4 and schema in Fig 3.9.
2	Predicts 360 saliency volumes using the dataset form the 360 Salient Challenge. Described in 3.5 and schema in Fig 3.11.

Table 4.1: Description of the models and their ids.

4.3 Training and validation losses

One of the metrics that helped us evaluate the how well our models were learning was the training and validation losses. It helped us to monitor the convergence of the models and guess if they were overfitting or underfitting the dataset.

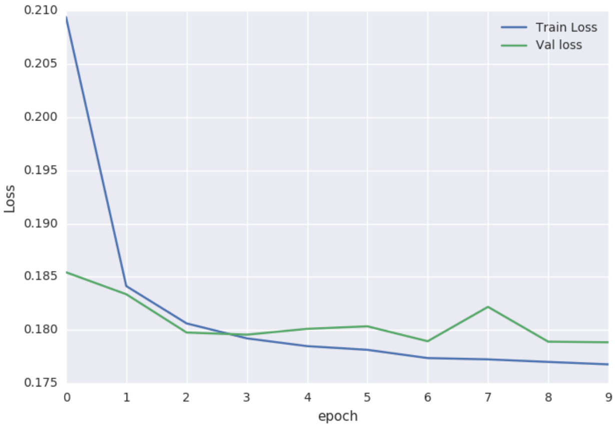


Figure 4.1: On the left, we have the learning curves of model 0. On the right, we have an example of an output.

In the case of model 0, we observe that the model learns very fast in the first two epoch to always predict fixations in the center (see Figure 3.6) and then stops learning. The loss function gets stuck at a very high value. This was caused by the model learning to produce fixations at the center of the image.

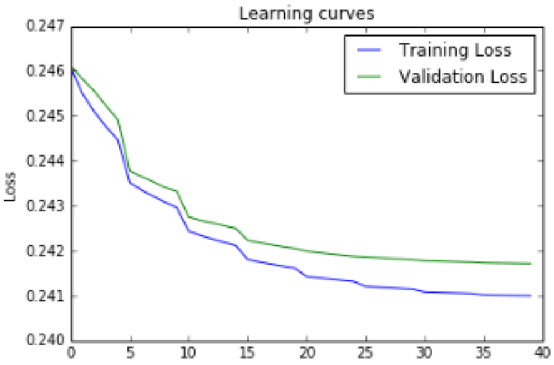


Figure 4.2: Learning curves of model 1.

With model 2, the learning rate was normalized by half each five epoch. This effect is notable and improves convergence. We also observe that the validation loss becomes flat in the end while the training loss slowly decreases. This suggests that the model might have started to overfit the training data.

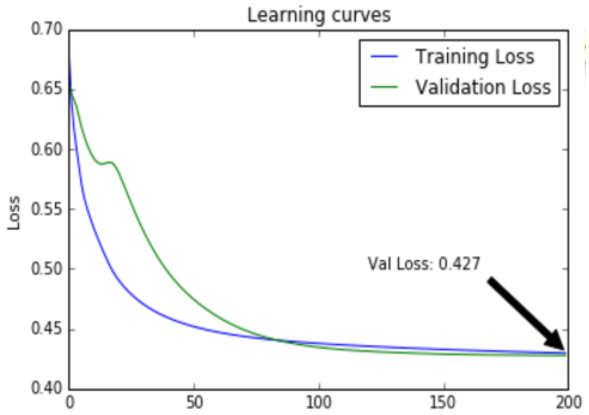


Figure 4.3: Learning curves of model 2.

Althought the dataset of model 3 is not very large, we don't observe overfitting. The network just stops learning in the end.

4.4 Quantitative results

4.4.1 Creating a baseline

At the time being, the field of scanpath prediction doesn't have a clear baseline that we can compare against. Therefore, we had to create a baseline. Using the metric that the Salient 360 challenge provided, we decided to compute the accuracies of: 1) random scanpaths, 2) the ground truth scanpaths, 3) scanpaths sampled from a ground truth saliency map, and 4) scanpaths sampled from a ground truth saliency volumes.

We observe that with completely random scanpaths get an accuracy of **4.94**[2]. In contrast, if we compare the ground truth scanpaths with themselves, we obtain an accuracy of **1.2e-8**, which approaches zero.

Sampling scanpaths from the ground truth saliency map has an accuracy of **1.89**. The accuracy is improved up to **1.79** if we sample the ground truth saliency volume.

The obtained baseline is coherent with our expectations, and it is presented in Table 4.2.

4.4.2 Model evaluation

Once we had a model capable of predicting saliency volumes, we evaluated different sampling strategies to generate scanpaths. An overview of these strategies can be found in *3.4.3 Sampling strategies to generate scanpaths from saliency volumes*.

The results are presented in Table 4.2, where we can see that the best performing sampling strategy limits the distance between fixations.

[2] Less is better

	Jarodzka↓
Random scanpaths	4.94
(1) Naive sampling strategy	3.45
(3) Avoiding fixating on same places	2.82
(2) Limiting distance between fixations	**2.27**
Sampling with (2) from ground truth saliency map	1.89
Sampling with (2) from ground truth saliency volume	1.79
Ground truth scanpaths	1.2e-8

Table 4.2: Comparison between the three considered spatial sampling strategies. Lower is better.

4.5 Qualitative results

Below we will present some qualitative examples of the results obtained by our model. These results are from the test set. We will also compare the different sampling strategies.

These examples have been chosen randomly from the results in order to obtain a sample as representative as possible of the whole dataset.

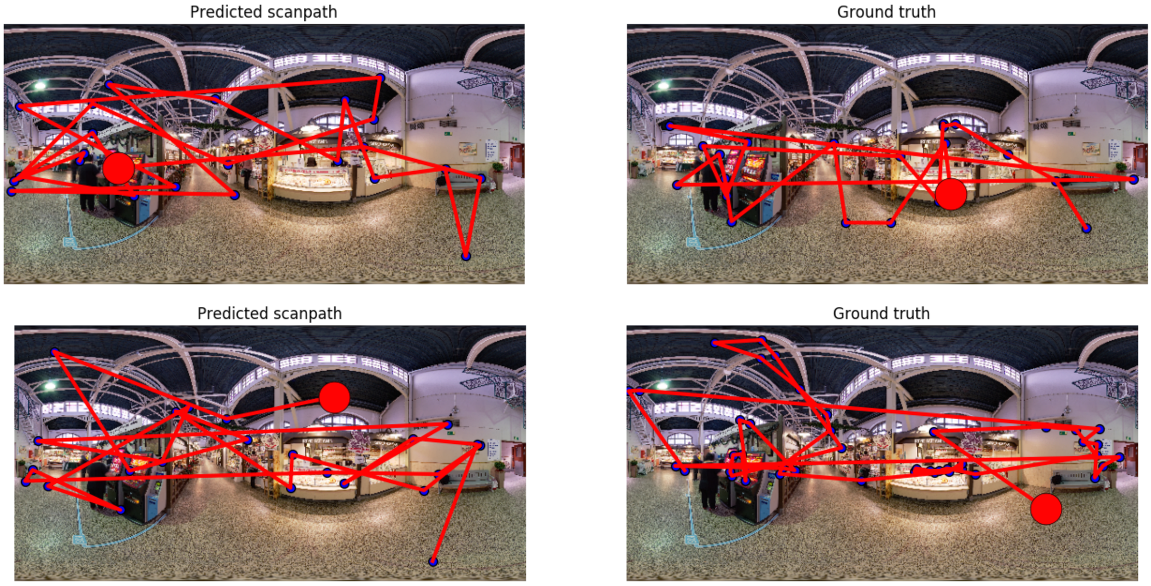


Figure 4.4: Examples of predicted and ground truth scanpaths on omi-directional photos from the 360Salient! dataset.

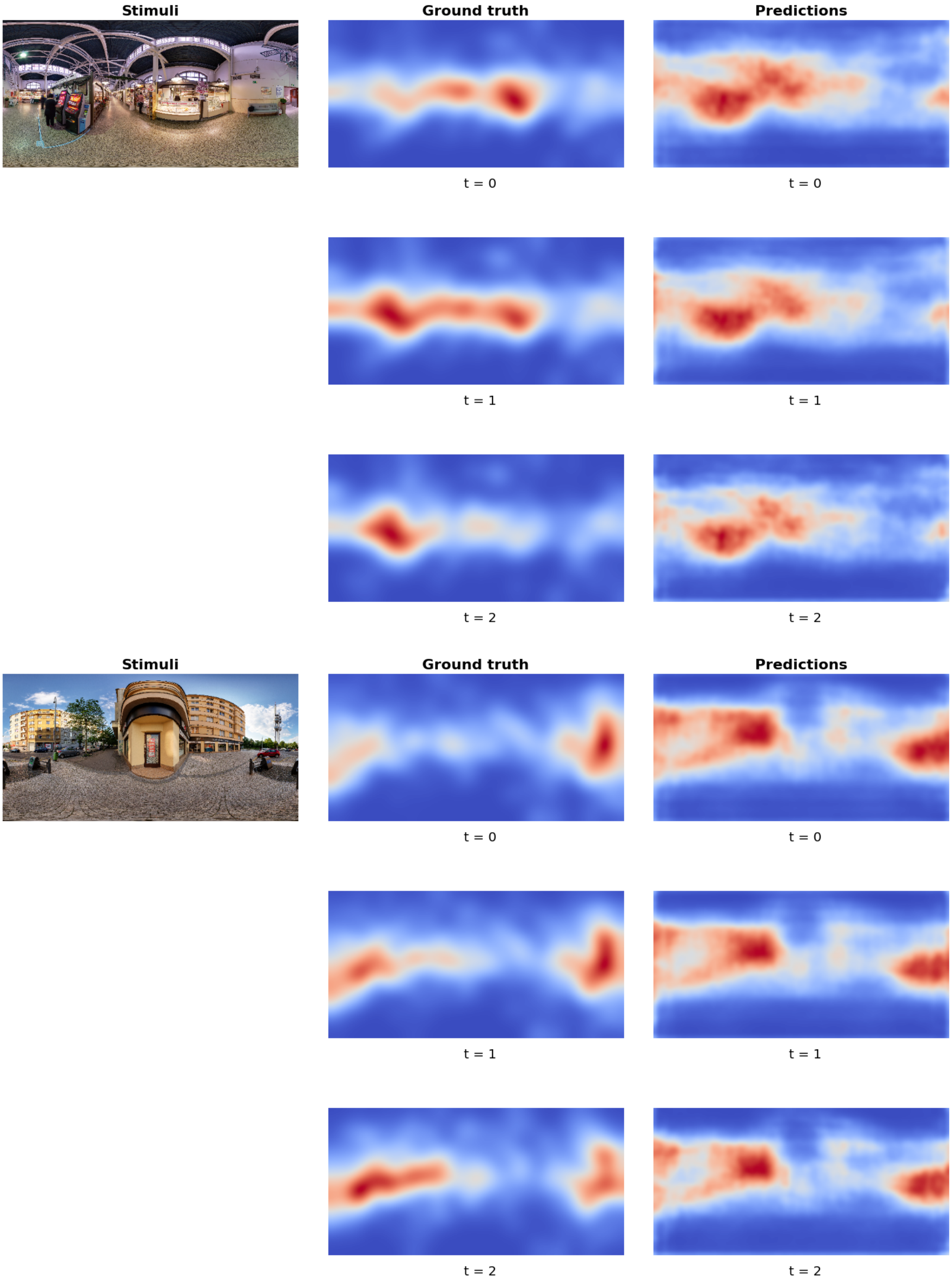


Figure 4.5: Examples of predicted and ground truth saliency volumes using the 360Salient! dataset.

Chapter 5

Budget

This project has been developed using the resources provided by the Insight Center for Data Analytics in the Dublin City University. Thus, this cost will not be reflected in the budget.

The main costs of the project are due to the salaries of the researchers that worked on it. I will consider that my salary is equivalent to the one of a junior engineer, and the salaries of my supervisors are equivalent to the ones of senior engineers.

Regarding the software that was used in the project, all of it was open source and it does not have costs associated. I will consider that the total duration of the project was 25 weeks, as depicted in the Gantt diagram in Figure 1.3.

	Amount	**Wage/hour**	**Dedication**	**Total**
Junior engineer	1	8,00 €/h	30 h/week	6,000 €
Senior engineer	2	20,00 €/h	4 h/week	4,000 €
			Total	10,000 €

Table 5.1: Budget of the project

Chapter 6

Conclusions

In this work, we have studied the time dimension of visual attention. First, we present a model capable of predicting scanpaths on 360-degree images. Second, we introduce a novel temporal-aware saliency representation that is able to generate other standard representations such as scanpaths, saliency maps or temporally weighted saliency maps. Our experiments show that it is possible to obtain realistic scanpaths by sampling from saliency volumes, and the accuracy greatly depends on the sampling strategy.

We successfully submitted our model to the 360Salient! challenge from the ICME conference, and we have been notified by the organization that we are expected to recieve an award. Marc will be presenting our model at Hong Kong next July.

We have also encountered the following limitations to the generation of scanpaths from saliency volumes: 1) the probability of a fixation is not conditioned to previous fixations; 2) the length of the scanpaths and the duration of each fixation are treated as independent random variables. We have tried to address the first problem by using more complex sampling strategies. Nevertheless, these three parameters are not independently distributed and therefore our model is not able to accurately represent this relationship.

An obvious next step would be to generate realistic scanpaths using generative adversarial models. This work was also shared with the scientific community through a preprint paper in arXiv and the publication of the source code and trained models at `https://github.com/massens/saliency-360salient-2017`. In addition, we plan to submit this paper at the ICCV 2017 workshop on ego-centric perception, interaction and computing.

Finally, in the process of developing the project we have also made three small open source contributions to the scientific community[1][2][3].

[1] `https://github.com/massens/salnet-keras`

[2] `https://github.com/mattnedrich/MeanShift_py/pull/3`

[3] `https://github.com/massens/tidybot`

Chapter 7

Appendices

As appendices, we attach the paper that we have presented at the ICCV workshop on egocentric perception, interaction and computing.

Scan-path Prediction on 360 Degree Images using Saliency Volumes

Marc Assens and Xavier Giro-i-Nieto
Image Processing Group
Universitat Politecnica de Catalunya (UPC)
Barcelona, Catalonia/Spain
xavier.giro@upc.edu

Kevin McGuiness and Noel E. O'Connor
Insight Center for Data Analytics
Dublin City University
Dublin, Ireland
kevin.mcguinness@insight-centre.org

Abstract

We introduce a deep neural network for scan-path prediction trained on 360 degree images, and a temporal-aware novel representation of saliency information named saliency volume. The first part of the network consists of a model trained to generate saliency volumes, whose weights are learned by back-propagation computed from a binary cross entropy (BCE) loss over downsampled versions of the saliency volumes. Sampling strategies over these volumes are used to generate scan-paths over the 360 degree images. Our experiments show the advantages of using saliency volumes, and how they can be used for related tasks. Our source code and trained models available at `https://github.com/massens/saliency-360salient-2017`.

1. Motivation

Visual saliency prediction is a field in computer vision that aims to estimate the areas of an image that attract the attention of humans. This information can provide important clues to human image understanding. The data collected for this purpose are fixation points in an image, produced by a human observer that explores the image for a few seconds, and are traditionally collected with eye-trackers [30], mouse clicks [13] and webcams [15]. The fixations are usually aggregated and represented with a saliency map, a single channel image obtained by convolving a Gaussian kernel with each fixation. The result is a gray-scale heatmap that represents the probability of each pixel in an image being fixated by a human, and it is usually used as a soft-attention guide for other computer vision tasks.

Saliency maps as a representation only describe saliency information with respect to image locations. Some recent studies have arised the need for a representation of saliency information that is dependent on time, and expresses how salient regions change with time [3]. Understanding the order by which humans scan through the content of an image or video has also stirred interest in the industry [29], as it can help in relevant areas of computer vision such as rendering devices to improve the quality of VR/AR content. In particular, we focus on 360° images, which represent the amount of visual data available to a human in a given context. Although a huge number of algorithms have been developed over the last years to gauge visual attention in flat-2D images and videos, attention studies in 360 scenarios are absent.

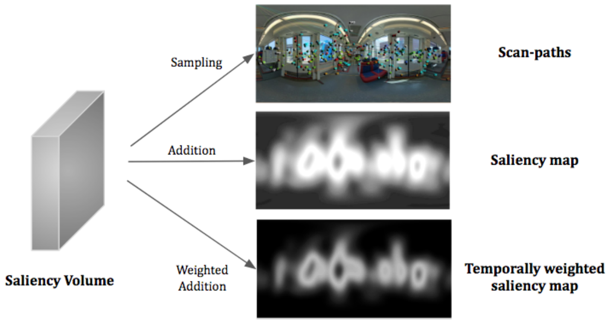


Figure 1: Scan-paths, saliency maps and temporally weighted saliency maps can be generated from a saliency volume.

This paper aims to formulate a model that is able to predict scan-paths in 360° images, and uses a novel temporal-aware saliency representation named saliency volume to accomplish the task.

$$\begin{aligned} Saliency\ map &= f(x, y) \\ Saliency\ volume &= f(x, y, t) \end{aligned} \quad (1)$$

2. Related Work

The first models for saliency prediction were biologically inspired and based on a bottom-up computational model that extracted low-level visual features such as intensity, color, orientation, texture and motion at multiple scales. Itti et al. [11] proposed a model that combines multiscale low-level

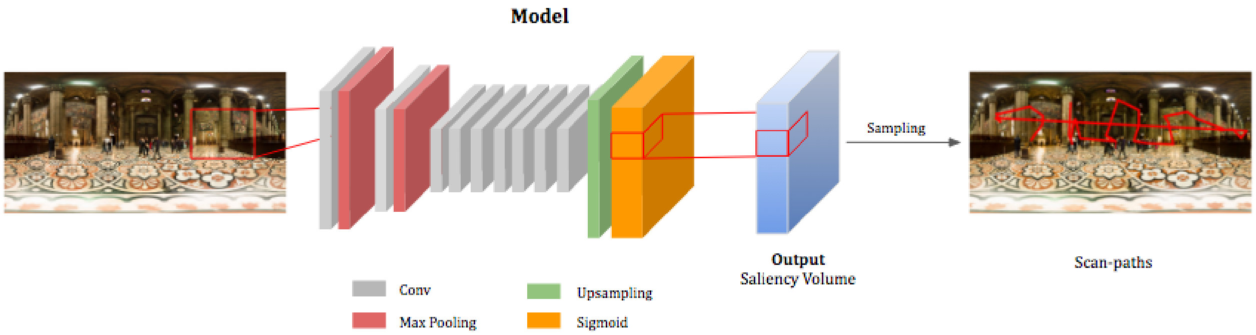


Figure 2: Overall architecture of the proposed scan-path estimation system.

features to create a saliency map. Harel et al. [8] presented a graph-based alternative that starts from low-level feature maps and creates Markov chains over various image maps, treating the equilibrium distribution over map locations as activation and saliency values.

Though this models did well qualitatively, the models had limited use because they frequently did not match actual human saccades from eye-tracking data. It seemed that humans not only base their attention on low-level features, but also on high-level semantics [3] (e.g., faces, humans, cars, etc.). Judd et al. introduced in [14] an approach that used low, mid and high-level image features to define salient locations. This features where used in combination with a linear support vector machine to train a saliency model. Borji [1] also combined low-level features with top-down cognitive visual features and learned a direct mapping to eye fixations using Regression, SVM and AdaBoost calssifiers.

Recently, the field of saliency prediction has made great progress due to advance of deep learning and it's applications on the task of Image Recognition [16] [26]. The advances suggest that this models are able to capture high-level features. As stated in [3], in March of 2016 there where six deep learning models among the top 10 results in the MIT300 saliency Benchmark [2]. In [25] Pan et al. compared shallow and deeper CNNs. Following the idea of modeling bottom-up and top-down features at the same time, Liu et al. [22] presented a multiresolution convolutional neural network (Mr-CNN) that combines predictions at different resolutions using a final logistic regression layer to predict a saliency map.

The enormous amount of training data necessary to train these netowrks makes them difficult to train directly for saliency prediction. With the objective of allowing saliency models to capture this high-level features, some authors have adapted well-known models with good performance in the task of Image Recognition. DeepGaze [17] achived state of the art performance by reusing the well-known AlexNet [16] pretrained on ImageNet [7] with a network on top that reads activations from the different layers of AlexNet. The output of the network is then blurred, center biased and converted to a probability distribution using a softmax. A second version called DeepGaze 2 [19] used features from VGG-19 [27] trained for image recognition. In this case, they did not fine-tune the network. Rather, some readout layers were trained on top of the VGG features to predict saliency with the SALICON dataset [13]. This results corroborated that deep features trained on object recognition provide a versatile feature space for performing related visual tasks.

In [28], Torralba et al. studied how the scene modules visual attention and discovered that the same objects recieve different attention depending on the scene where they appear (i.e. pedestrians are the most salient object in only 10% of the outdoor scene images, being less salient than many other objects. Tables and chairs are among the most salient objects in indoor scenes). With this insight, Liu et al. proposed DSCLRCN [21], a model based on CNNs that also incorporates global context and scene context using RNNs. Their experiments have obtained outstanding results in the MIT Saliency Benchmark.

Recently, there has been interest in finding appropiate loss functions. Huang et al. [10] made an interesting contribution by introducing loss functions based on metrics that are differentiable, such as NSS, CC, SIM and KL divergence to train a network (see [25] and [18]).

Other advances in deep learning such as generative adversarial training (GANs) and attentive mechanisms have also been applied to saliency prediction: Pan et al. recently introduced SalGAN [23], a deep network for saliency prediction trained with adversarial examples. As all other Generative Adversarial Networks, it is composed by two modules, a generator and a discriminator, which combine efforts to produce saliency maps. Cornia et al. presented in [6] a model that incorporates neural attentive mechanisms. The model includes a Convolutional LSTM that focuses on the most salient regions of the image to iteratively refine the predicted saliency map. Additionally, they tackle the center bias present in human eye fixations by learning a set of prior map generated by Gaussian functions.

2.1. Scanpath prediction

Unlike with the related task of saliency map prediciton, there hasn't been much progress in the task of scanpath prediciton over the last years. Cerf et al. [4] discovered that observers, even when not instructed to look for anything particular, fixate on a human face with a probability of over 80% within their first two fixations. Furthermore, they exhibit more similar scanpaths when faces are present. Recently, Hu et al. [9] have introduced a model capable of selecting relevant areas of a 360° video and deciding in which direction should a human observer look at each frame. An object detector is used to propose candidate objects of interest and a RNN selects the main object at each frame.

3. Architecture

The architecture of the presented model is based on a deep convolutional neural network (DCNN), that predicts a saliency volume for a given input image. This section provides detail on the structure of the network, the loss function, and the strategy used to generate scanpaths from saliency volumes.

3.1. Saliency Volumes

Saliency volumes aim to be a suitable representation of spatial and temporal saliency information for images. They have three axes that represent the width and height of the image, and the temporal dimension.

Saliency volumes are generated from information already available in current fixation datasets. First, the timestamps of the fixations are quantized. The length of the time axis is determined by the longest timestamp and the quantization step. Second, a binary volume is created by placing '1' on the fixation points and '0' on the rest of the positions. Third, a multivariate Gaussian kernel is convolved with the volume to generate the saliency volume. The values of each temporal slice are normalized, converting the slice into a probability map that represents the probability of each pixel being fixated by a user at each timestep.

Figure 5 shows how saliency volumes are a meta-representation of saliency information and other saliency representations can be extracted from them. Saliency maps can be generated by performing an addition operation across all the temporal slices of the volume, and normalizing the values to ensure they add to one. A similar representation are *temporally weighted saliency maps*, which are generated by performing a weighted addition operation of all the temporal slices. Finally, scan-paths can also be extracted by sampling fixation points from the temporal slices. Sampling strategies that aim to generate realistic scan-paths are will be discussed in the 4.Experiments section.

3.2. Neural network

We propose a network that adapts the filters learned to predict flat saliency maps to predict saliency volumes. Figure 6 illustrates the architecture of the convolutional neural network, composed of 10 layers and a total of 25.8 million parameters. Each convolutional layer is followed by a rectified linear unit non-linearity (ReLU). Excluding the last layer, the architecture follows the proposal of SalNet [24], whose first three layers were at the same time extracted from the VGG-16 model [5] trained for image classification.

Our network was designed considering the amount of training data available. Different strategies where introduced to prevent overfitting. Firstly, the model was previously trained on the similar task of saliency map prediction, and the obtained weights were fine-tunned for the task of saliency volume prediction. Secondly, the input images where resized to $[300 \times 600]$, a much smaller dimension than their initial size $[3000 \times 6000]$. The last layer of the network outputs a volume with size $[12 \times 300 \times 600]$, with three axis that represent time, and height and width of the image.

3.3. Scan-path sampling

The generation of scan-paths from the saliency volumes requires determining: 1) number of fixations of each scanpath; 2) the duration in seconds of each fixation; and 3) the location of each fixation point. The first two values were sampled from their probability distributions learned from the training data. The location of each fixation point was also generated by sampling, this time from the corresponding temporal slice from the predicted saliency volume. Different strategies were explored for this purpose, presented together with their performance in Section 5.

4. Training

We trained our network on 36 images of the 40 training images provided by the Salient360 dataset [29], leaving aside 4 images for validation. We normalized the values of the saliency volumes to be in the interval of [0, 1]. Both the input images and the saliency volumes were downsampled to 600×300 prior to training. The saliency volumes where generated from fixations using a multivariate Gaussian kernel with bandwidths $[4, 20, 20]$ (time, height, width).

The network was trained using stochastic gradient descent with Cross Entropy loss using a batch size of 1 image during 90 epoch. During training, results on the validation set were tracked to monitor convergence and overfitting problems. The L^2 weight regularizer (weight decay) was used to avoid overfitting. Our network took approximately two hours to train on a NVIDIA GTX Titan X GPU running the Keras

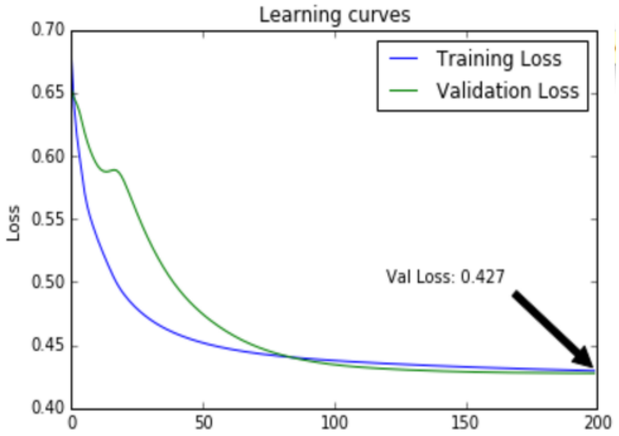


Figure 3: Training curves for our model with Binary Cross entropy loss.

framework with Theano backend. The learning rate was set to $\alpha = 0.001$ during all the training.

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{j=1}^{N} S_j \log(\hat{S}_j) + (1 - S_j)\log(1 - \hat{S}_j). \tag{2}$$

Due to the small size of the training dataset, we performed transfer learning to initialize the weights of the network using related tasks. First, the network was trained to predict saliency maps using the SALICON dataset [10] using the same architecture of SalNet [24]. Then, the network was trained to predict saliency volumes generated from the iSUN dataset [31] that contains 6000 training images. Finally, the network was fine-tuned using the images provided by the 360° Salient challenge [29].

5. Experiments

The presented model for scan-path prediction was assessed and compared from different perspectives. First, we assess the impact of different sampling strategies to generate scan-paths from saliency volumes. Finally, we show quantitative performance results of the model.

5.1. Sampling strategies

The sampling of the number of fixations and their durations were drawn from the data distribution on the training set plotted in Figure 4.

Regarding the spatial location of the fixation points, three different strategies were explored. The simplest approach (1) consists on taking one fixation for each temporal slice of the saliency volume. Through qualitative observation we noticed that scan-paths were unrealistic, as the probability of each fixation is not conditioned to previous fixations. A more elaborated sampling strategy (2) consists on forcing fixations to be closer to their respective previous fixation. This is accomplished by multiplying a temporal slice (probability map) of the saliency volume with a Gaussian kernel centered at the previous fixation point. This suppresses the probability of positions that are far from the previous fixation point. The third sampling strategy (3) we assessed consisted on suppressing the area around all the previous fixations using Gaussian kernels. As shown in Table 1, we observe that the best performing model is the one using the sampling strategy (2).

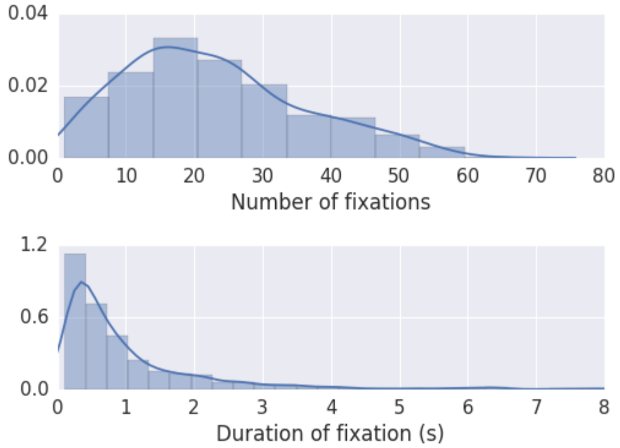


Figure 4: Probability distribution of the number of fixations per scan-paths (top) and duration of each fixation (bottom).

5.2. Results

Scan-path prediction evaluation has received attention lately and it is a very active field of research [20][12]. For this work, we have used a metric proposed by the Salient 360 Challenge [29] organization that compares the similarity of 40 generated scan-paths with the ground truth scan-paths. The similarity metric used is the Jarodzka algorithm [12], where the similarity criteria was slightly modified to use equirectangular distances in 360 instead of euclidean distances. Also, the generated and ground truth scanpaths are matched 1 to 1 using the Hungarian optimizer to get the least possible final cost. Table 1 exposes the performance results of our model using different sampling strategies (discussed in the section below). Due to the lack of a baseline from other scan-path prediction models, we have compared our results with the accuracy that would obtain a model that outputs random fixations, and a model that outputs the ground truth fixations.

	Jarodzka↓
Random scanpaths	4.94
(1) Naive sampling strategy	3.45
(3) Combined sampling strategy	2.82
(2) Limiting distance between fixations	2.41
Ground truth	1.2e-8

Table 1: Comparison between the three considered spatial sampling strategies.

6. Conclusions

In this work we have presented a model capable of predicting scan-paths on 360° images. We have also introduced a novel temporal-aware saliency representation that is able to generate other standard representations such as scan-paths, saliency maps or temporally weighted saliency maps. Our experiments show that it is possible to obtain realistic scan-paths by sampling from saliency volumes, and the accuracy greatly depends on the sampling strategy.

We have also found the following limitations to the generation of scan-paths from saliency volumes: 1) the probability of a fixation is not conditioned to previous fixations; 2) the length of the scan-paths and the duration of each fixation are treated as independent random variables. We have tried to address the first problem by using more complex sampling strategies. Nevertheless, this three parameters are not independently distributed and therefore our model is not able to accurately represent this relationship.

Our results can be reproduced with the source code and trained models available at `https://github.com/massens/saliency-360salient-2017`.

7. Acknowledgments

The Image Processing Group at UPC is supported by the project TEC2013-43935-R and TEC2016-75976-R, funded by the Spanish Ministerio de Economia y Competitividad and the European Regional Development Fund (ERDF). The Image Processing Group at UPC is a SGR14 Consolidated Research Group recognized and sponsored by the Catalan Government (Generalitat de Catalunya) through its AGAUR office. This material is based upon works supported by Science Foundation Ireland under Grant No 15/SIRG/3283. We gratefully acknowledge the support of NVIDIA Corporation with the donation of a GeForce GTX Titan X used in this work.

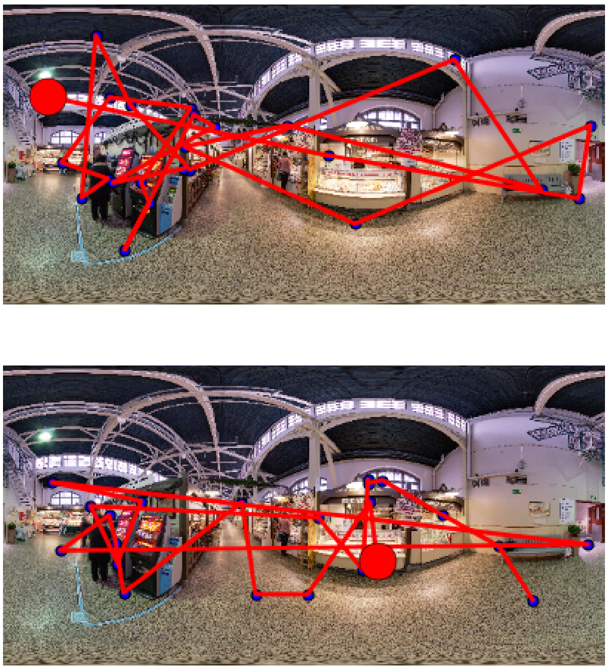


Figure 5: The top image shows a predicted scanpath, sampled from a prdicted saliency volume. The image at the bottom shows a single ground truth scanpath.

References

[1] A. Borji. Boosting bottom-up and top-down visual features for saliency estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 438–445. IEEE, 2012.

[2] Z. Bylinskii, T. Judd, A. Ali Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. MIT saliency benchmark. http://saliency.mit.edu/.

[3] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand. Where should saliency models look next? In *European Conference on Computer Vision*, pages 809–824. Springer, 2016.

[4] M. Cerf, J. Harel, W. Einhäuser, and C. Koch. Predicting human gaze using low-level saliency combined with face detection. In *Advances in neural information processing systems*, pages 241–248, 2008.

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[6] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *arXiv preprint arXiv:1611.09571*, 2016.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[8] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2007.

[9] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for

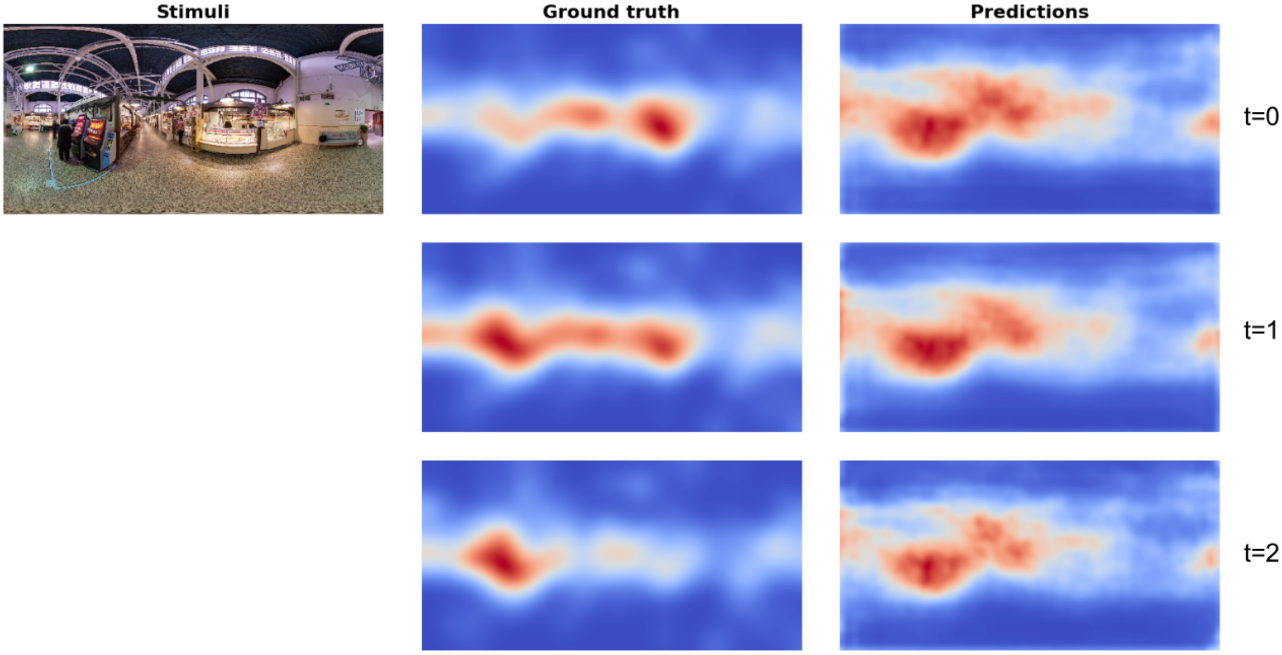


Figure 6: The images shown above show the predicted and ground truth saliency volumes for a given stimulus. For each saliency volume, three temporal slices are shown.

piloting through 360 {\deg} sports video. *arXiv preprint arXiv:1705.01759*, 2017.

[10] X. Huang, C. Shen, X. Boix, and Q. Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[11] L. Itti, C. Koch, and E. 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.

[12] H. Jarodzka, K. Holmqvist, and M. Nyström. A vector-based, multidimensional scanpath similarity measure. In *Proceedings of the 2010 symposium on eye-tracking research & applications*, pages 211–218. ACM, 2010.

[13] M. Jiang, S. Huang, J. Duan, and Q. Zhao. Salicon: Saliency in context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[14] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th international conference on*, pages 2106–2113. IEEE, 2009.

[15] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

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

[18] M. Kümmerer, T. S. Wallis, and M. Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112(52):16054–16059, 2015.

[19] M. Kümmerer, T. S. Wallis, and M. Bethge. DeepGaze II: Reading fixations from deep features trained on object recognition. *ArXiv preprint:1610.01563*, 2016.

[20] O. Le Meur and T. Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013.

[21] N. Liu and J. Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *arXiv preprint arXiv:1610.01708*, 2016.

[22] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu. Predicting eye fixations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 362–370, 2015.

[23] J. Pan, C. Canton, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

[24] J. Pan, E. Sayrol, X. Giró-i Nieto, K. McGuinness, and N. E. O'Connor. Shallow and deep convolutional networks for saliency prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit. Saliency and human fixations: State-of-the-art and study

of comparison metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 1153–1160, 2013.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] A. Torralba, A. Oliva, M. S. Castelhano, and J. 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.

[29] T. University of Nantes. Salient360: Visual attention modeling for 360 images grand challenge, 2017.

[30] N. Wilming, S. Onat, J. P. Ossandón, A. Açık, T. C. Kietzmann, K. Kaspar, R. R. Gameiro, A. Vormberg, and P. König. An extensive dataset of eye movements during viewing of complex images. *Scientific Data*, 4:160126, 2017.

[31] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

Bibliography

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

[2] Ali Borji. Boosting bottom-up and top-down visual features for saliency estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 438–445. IEEE, 2012.

[3] Zoya Bylinskii, Tilke Judd, Ali Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. MIT saliency benchmark. http://saliency.mit.edu/.

[4] 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.

[5] Moran Cerf, Jonathan Harel, Wolfgang Einhäuser, and Christof Koch. Predicting human gaze using low-level saliency combined with face detection. In *Advances in neural information processing systems*, pages 241–248, 2008.

[6] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *arXiv preprint arXiv:1611.09571*, 2016.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[8] C Ehmke and C Wilson. Identifying web usability issues from eye-tracking data. *People and Computers XXI–HCI... but not as we know it: Proceedings of HCI*, 2007.

[9] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2007.

[10] Hou-Ning Hu, Yen-Chen Lin, Ming-Yu Liu, Hsien-Tzu Cheng, Yung-Ju Chang, and Min Sun. Deep 360 pilot: Learning a deep agent for piloting through 360 {\deg} sports video. *arXiv preprint arXiv:1705.01759*, 2017.

[11] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 262–270, 2015.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[13] 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.

[14] Halszka Jarodzka, Kenneth Holmqvist, and Marcus Nyström. A vector-based, multidimensional scanpath similarity measure. In *Proceedings of the 2010 symposium on eye-tracking research & applications*, pages 211–218. ACM, 2010.

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

[16] Sheree Josephson and Michael E Holmes. Visual attention to repeated internet images: testing the scanpath theory on the world wide web. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 43–49. ACM, 2002.

[17] Sheree Josephson and Michael E Holmes. Clutter or content?: how on-screen enhancements affect how tv viewers scan and what they learn. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 155–162. ACM, 2006.

[18] 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.

[19] 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.

[20] 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.

[21] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112(52):16054–16059, 2015.

[22] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Deepgaze ii: Reading fixations from deep features trained on object recognition. *arXiv preprint arXiv:1610.01563*, 2016.

[23] Olivier Le Meur and Thierry Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013.

[24] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[25] Nian Liu and Junwei Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *arXiv preprint arXiv:1610.01708*, 2016.

[26] Nian Liu, Junwei Han, Dingwen Zhang, Shifeng Wen, and Tianming Liu. Predicting eye fixations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 362–370, 2015.

[27] S Mannan, KH Ruddock, and DS Wooding. Automatic control of saccadic eye movements made in visual inspection of briefly presented 2-d images. *Spatial vision*, 9(3):363–386, 1995.

[28] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[29] 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.

[30] 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.

[31] Yashas Rai, Jesus Gutiérrez, and Patrick Le Callet. A dataset of head and eye movements for omni-directional images. In *Proceedings of the 8th International Conference on Multimedia Systems*. ACM, 2017.

[32] 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 *Proceedings of the IEEE international conference on computer vision*, pages 1153–1160, 2013.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

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

[35] 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.

[36] Ekaterini Tzanidou, Shailey Minocha, and Marian Petre. Applying eye tracking for usability evaluations of e-commerce sites. In *workshop on'Commercial Uses of Eye tracking'held at the 19th British HCI Group Annual Conference, Napier University, Edinburgh*, 2005.

[37] HUMPHREY K. UNDERWOOD, G. and T. FOULSHAM. Hci and usability for education and work. pages 125–144, 2008.

[38] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015.

[39] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.