UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Visual Saliency Prediction using Deep learning Techniques

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

By

Junting Pan

In partial fulfilment

of the requirements for the degree in

TELECOMUNICATION ENGINEERING

Advisor: Xavier Giró i Nieto

Barcelona, July 2015

Abstract

A saliency map is a model that predicts eye fixations on a visual scene. In other words, it is the prediction of saliency areas in images has been traditionally addressed with hand crafted features inspired on neuroscience principles. This work however addresses the problem with a completely data-driven approach by training a convolutional network.

The recent publication of large datasets of saliency prediction has provided enough data to train a not very deep network architecture which is both fast and accurate. In our system, named JuntingNet, the learning process is formulated as a minimization of a loss function that measures the Euclidean distance of the predicted saliency map with the provided ground truth.

The convolutional network developed in this work, named JuntingNet, won the CVPR Large-scale Scene UNderstanding (LSUN) 2015 challenge on saliency prediction with a superior performance in all considered metrics.

Resum

Un mapa de prominència és un model que prediu els punts de fixació dels ulls en una escena visual. Tradicionalment, aquest problema s'ha resolt amb descriptors visuals dissenyats manualment inspirats en principis de la neurociència. Aquest treball, en canvi, es planteja el problema desde d'un punt de vista purament basat en dades, que entrenen una xarxa convolucional.

La recent publicació d'un gran volum de mapes de prominència ha fet possible l'entrenament d'una xarxa convolucional no gaire profunda. A la xarxa dissenyada, el procés d'aprenentatge es formula com la minimització d'una funció de cost que mesura la distància euclidiana entre el mapa predit i la seva veritat terreny.

La xarxa convolucional desenvolupada en aquest treball, anomenada JuntingNet, es va imposar en la categoria de predicció de prominència en el concurs CVPR Large-scale Scene UNderstanding (LSUN) 2015, amb uns resultats clarament superiors en totes les mètriques considereades.

Resumen

Un mapa de prominencia es un modelo que explica los puntos de fijación de los ojos en una escena visual. Tradicionalmente este problema se ha resuelto con descriptores visuales diseñados manualmente inspirados principios de la neurociencia. Este trabajo, en cambio, se plantea el problema desde un punto de vista puramente basado en datos, que entrenan una red convolucional.

La reciente publicación de gran volumen de mapas de prominencia ha hecho posible el entrenamiento de una red convolucional no muy profunda. En la red diseñada, el proceso de aprendizaje se formula como la minimización de una función de coste que mide la distancia euclidiana entre el mapa de prominencia y su verdad terreno.

La red covolucional desarrollado en este trabajo, llamada JuntingNet, se impuso en la categoría de predicción de prominencia en el concurso CVPR Large-scale Scene UNderstanding (LSUN) 2015, con unos resultados claramente superiores en todas las métricas consideradas.

Acknowledgements

It is my proud privilege to release the feeling of my gratitude to several persons who helped me to conduct this project work.

First of all, I would like to express my deep sense of gratitude to my project advisor Prof. Xavier Giró i Nieto for giving me the chance to take part in such an attractive project and challenge, and his valuable guidance, keen interest and encouragement at various stages of my project development period.

I am very much thankful to Carlos Segura and Carles Fernández whose suggestion and inspiration have contributed to the evolution of my ideas on the project.

I also thank all my colleagues in the DeepGPI group who have more or less contributed to the development of this project.

I thank profusely all the software experts from GPI: Albert Gil and Josep Pujal, who always solve all technical problems at earliest time possible.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GeoForce GTX 980 used in this work.

Revision history and approval record

Revision	Date	Purpose
0	27/06/2015	Document creation
1	08/07/2015	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Junting Pan	junting.pa@gmail.com
Xavier Giró Nieto	xavi.giro@upc.edu

Written by:		Reviewed and approved by:	
Date	27/06/2015	Date	08/07/2015
Name	Junting Pan	Name	Xavier Giró Nieto
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables:

1. Introduction

Recently, visual saliency has received considerable growing attention across many disciplines such as neurobiology, image processing, artificial intelligence and computer vision. Human visual system process only parts of an image in detail, with only a limited processing of images areas out of the focus, all of these are due to the reaction time of observation and signal transmission time along the neurons and cells. These and that is why saliency models can estimate the user attention with applications to marketing, image compression, social studies...

1.1. Motivation and contributions

This work presents an end-to-end convolutional network (ConvNet) for saliency prediction. Our objective is to compute saliency maps that represent the probability of visual attention. This problem has been traditionally addressed with hand-crafted features inspired by neurology studies. In our case we have adopted a completely data-driven approach, training a model with a large amount of annotated data.

ConvNet is a popular architecture in the field of deep learning and has been widely explored for visual pattern recognition, ranging from a global scale image classification to a more local object detection or semantic segmentation. The hierarchy of layers of ConvNet is also inspired by biological models and actually recent works have pointed at a relation between the activities of certain areas in the brain with hierarchy of layers in the ConvNet [1]. Provided with enough training data, ConvNet shows impressive results, often outperforming other hand-crafted methods. In many popular works, the output of the ConvNet is a discrete label associated to a certain semantic class. The saliency prediction problem, though, addresses the problem of a continuous range of values that estimate the probability of a human fixation on a pixel. These values present a spatial coherence and smooth transition that this work addresses by using the ConvNet as a regression solver, instead of a classifier.

The training of a convolutional network requires a large amount of annotated data that provides a rich description of the problem. Our work has benefited from the recent publication of two datasets: iSun [21] and SALICON [12]. These datasets propose two different approaches for saliency prediction. While iSun was generated with an eye-tracker to annotate the gaze fixations, the SALICON dataset was built by asking humans to click on the most salient points on the image. The different nature of the saliency maps of the two datasets can be seen in Figure 1. The large size of these datasets has provided for the first time the possibility of training a ConvNet.

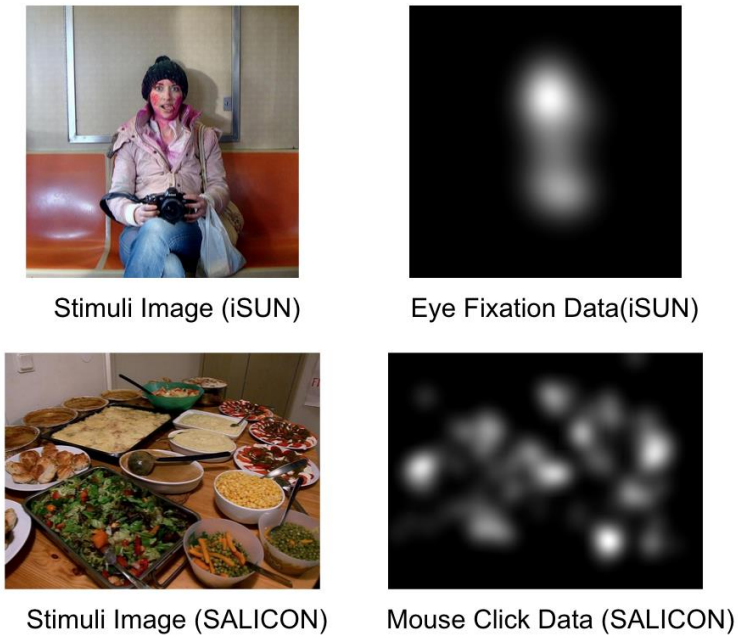

Figure 1 Images (right) and saliency maps (left) from the iSUN

Our main contribution has been the design of an end-to-end ConvNet for saliency prediction, the first one from this type, up to the author's knowledge. The network, called JuntingNet, has proved its superior performance in the Large-scale Scene UNderstanding (LSUN) challenge 2015 [23].

1.2. Requirements and specifications

Project requirements:

- Development of software that is able to predict the saliency map from its natural image.
- Comparative research of the state-of-art saliency prediction models
- Submitted of the results to scientific benchmarks such as the LSUN challenge and the MIT benchmark.
- In case of a satisfactory performance in the benchmarks, contribute to the scientific dissemination of the work.

Project specifications:

- The software is developed with Python programing language.
- Python deep learning library (Theano) is used.
- The Convolutional Neural Networks model is trained on GPUs in order to reduce the training time.

1.3. Work plan and Gantt diagram

<table>
<tr><td>Project proposal and work plan</td><td colspan="2">WP ref.: (WP1)</td></tr>
<tr><td>Documentation</td><td colspan="2">Sheet 1 of 7</td></tr>
<tr><td rowspan="2">Project description and organization. Gantt diagram.</td><td colspan="2">Planned start date: 25/02/2015
Planned end date: 06/03/2015</td></tr>
<tr><td colspan="2">Start event: 25/02/2015
End event: 11/03/2015</td></tr>
<tr><td>T1: Project description and planning
T2: Project plan redaction
T3: Project plan revision
T4: Project approval</td><td>Deliverables:
Project plan</td><td>Dates:</td></tr>
</table>

<table>
<tr><td>Analysis of state-of-art model and alternatives</td><td colspan="2">WP ref.: (WP2)</td></tr>
<tr><td>Documentation</td><td colspan="2">Sheet 2 of 7</td></tr>
<tr><td rowspan="2">Study and research of the state-of-art model about saliency prediction , visualization inside the networks and frameworks for development</td><td colspan="2">Planned start date: 27/02/2015
Planned end date: 30/03/2015</td></tr>
<tr><td colspan="2">Start event:27/02/015
End event:25/03/2015</td></tr>
<tr><td>T1: Stanford Course: Convolutional Neural Networks for Visual Recognition
T2: Lecture: "Visualizing and Understanding Convolutional Networks"
T3: Lecture: "Deep Inside Convolutional Networks: Visualizing Image Classification Models and Saliency maps"
T4: Caffe tutorial: a deep learning framework.</td><td>Deliverables:
A review of a paper on the BitSearch blog</td><td>Dates:</td></tr>
</table>

Analysis of the trained Convolutional Networks available on GPI's server.	WP ref.: (WP3)	
SW	Sheet 3 of 7	
Analyze the Convolutional Networks available on the server in order to decide the architecture for the saliency prediction. At the same time decide which frameworks or library will be use for the development.	Planned start date: 05/03/2015 Planned end date: 18/03/2015	
	Start event: 05/03/2015 End event:18/03/2015	
T1: Landing and getting use the GPI's work environment. T2: Comparative research between T3: Implementation of a prototype using a trained CNN. T4: Decisions about software development.	Deliverables: Prototype	Dates:

Program development	WP ref.: (WP4)	
SW	Sheet 4 of 7	
Design and implementation of the CNN with the chosen framework. Train the networks with the data sets, Cross validation and fine tuning to improve its performance and to avoid over fitting.	Planned start date: 05/03/2015 Planned end date: 30/04/2015	
	Start event:05/03/2015 End event:30/04/2015	
T1: Download training data sets T2: Setting up and getting started with Python T3: Installation of libraries and frameworks. T4: Design of the convolutional networks T5: Training with MIT saliency benchmark data sets. T6: Cross validation and fine tuning of the network.	Deliverables: SW	Dates:

Critical Review	WP ref.: (WP5)	
Evaluation	Sheet 5 of 7	
Project progress and reorganization of the project plan	Planned start date: 16/04/2015 Planned end date: 22/04/2015	
	Start event:16/04/2015 End event:23/04/2015	
T1: Review task progress, progress schedule and actual work T2: Restructure the work plan and decide the final development plan T3: Critical Design Review redaction	Deliverables: Critical Design Review	Dates:

Performance evaluation and decision for the LSUN challenge	WP ref.: (WP6)	
SW	Sheet 6 of 7	
Evaluate the networks that we have developed, with MIT benchmark data sets. Retrain the networks provide from the LSUN challenge.	Planned start date: 24/04/2015 Planned end date: 29/05/2015	
	Start event:24/05/2015 End event:23/06/2015	
T1: Performance evaluation using MIT benchmark data sets. T2: Retrain the CNN with data sets Provide from the LSUN challenge. T3: Cross validation and fine tuning of the network. T4: Submit the code to the LSUN challenge. T5: Write a scientific paper and/or poster to support the submission.	Deliverables: SW	Dates:

<table>
<tr><td>Writing the thesis and preparation for the oral presentation</td><td colspan="2">WP ref.: (WP7)</td></tr>
<tr><td>Documentation</td><td colspan="2">Sheet 7 of 7</td></tr>
<tr><td rowspan="2">Write thesis’s final report, and prepare slides for the oral defense.</td><td colspan="2">Planned start date:
26/06/2015
Planned end date:
15/07/2015</td></tr>
<tr><td colspan="2">Start event:25/06/2015
End event:20/07/2015</td></tr>
<tr><td>T1: Final Report redaction
T2: Final Report revision
T3: Final Report approval
T4: Prepare oral presentation
T5: Oral defense.
T6: Publish the source code and necessary data to reproduce the work.</td><td>Deliverables:
Final Report</td><td>Dates:</td></tr>
</table>

Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
WP1	T4	Project plan approval	Work Plan	11/03/2015
WP3	T2	Comparative research of libraries.	Research report	4th week
WP4	T4	Design of the networks	CNN structure	25/04/2015
WP4	T5	Training with MIT saliency benchmark data sets.	Trained Network	30/04/2015
WP4	T6	Cross validation and fine tuning.	Fine-tuned network for the MIT data sets.	30/04/2015
WP5	T3	Critical Design Review redaction	Critical design review	22/04/2015
WP6	T4	Submission the code to the LSUN challenge	Code	04/06/2015
WP6	T5	Write a scientific paper and/or poster to support the submission.	Scientific paper on arXiv	06/07/2015
WP7	T3	Final Report approval	Final report	08/07/2015
WP7	T5	Oral defense	Oral defense	20/07/2015

Gantt Diagram

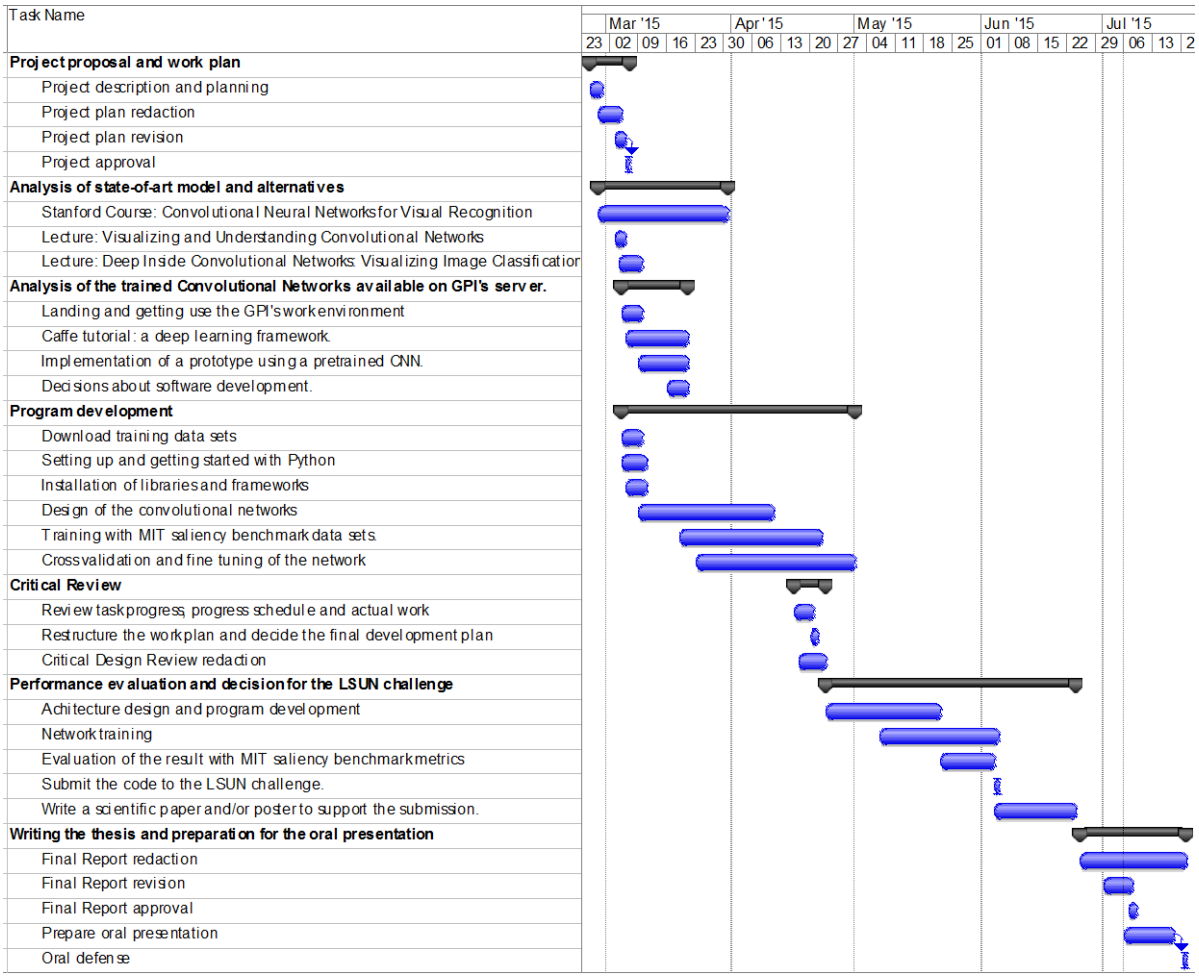

Figure 2 Gantt Diagram

1.4. Incidences

Since the submission of the critical review, there only have been some slight modifications:

1. The deadline for the LSUN challenge submission was extended until June 5th.
2. The scientific paper describing our work was submitted after the challenge at July 7th.

The other work packages remain the same, considering a slight change of dates due the two incidences previously explained.

2. State of the art of the technology used or applied in this thesis:

Our designed Convolutional Neural Network (ConvNet) provides the next natural step to two main trends in deep learning: using ConvNet for saliency prediction and training these networks by formulating an end-to-end problem.

2.1. Deep learning for saliency prediction

An early attempt of predicting saliency model with a ConvNet was the ensembles of Deep Networks (eDN) [19] depicted in Figure 3, which proposed an optimal blend of features from three different ConvNet layers who were finally combined with a simple linear classifier trained with positive (salient) or negative (non-salient) local regions.

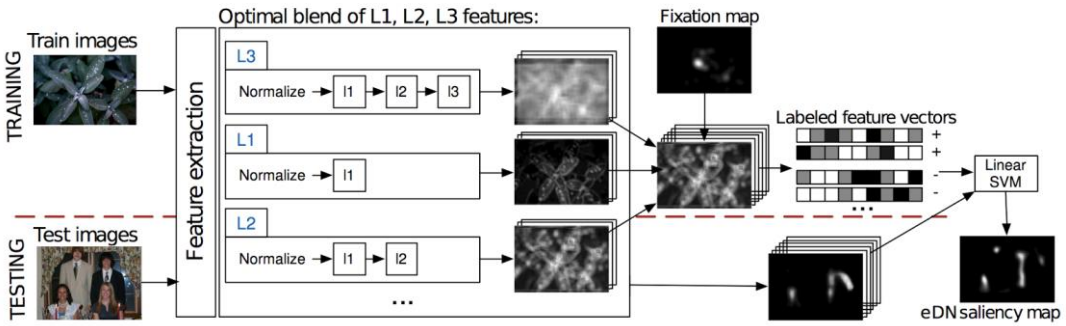

Figure 3 Schematic diagram of our pipeline of eDN [19] model.

This approach inspired DeepGaze [15] shown in **Error! Reference source not found.**, hich only combined features from different layers but, in this case, from a much deeper network. In particular, DeepGaze used the existing AlexNet ConvNet [14], which had been trained for an object classification task, not for saliency prediction. JuntingNet adopts a not very deep architecture as eDN, but it is end-to-end trained as a regression problem, avoiding the reuse of precomputed parameters from another task.

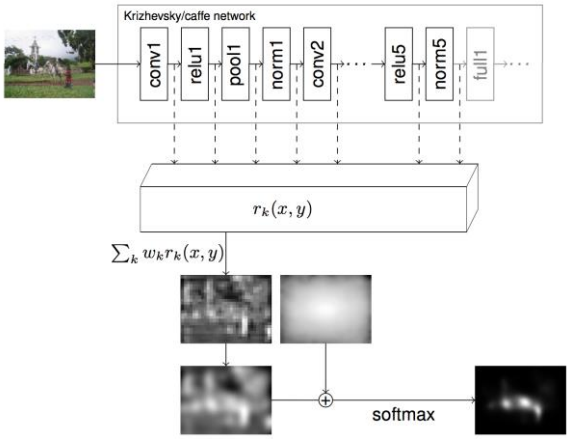

Figure 4 The model structure of Deep Gaze[15].

2.2. End-to-end semantic segmentation

Fully Convolutional Networks (FCNs) [17] addressed the semantic segmentation task which predicting the semantic label of every individual pixel in the image. This approach dramatically improved previous results on the challenging PASCAL VOC segmentation benchmark [6]. The idea of an end-to-end solution for a 2D problem as semantic segmentation was refined by DeepLab-CRF [5], where the spatial consistency of the predicted labels is checked with a Conditional Random Field (CRF), similarly to the hierarchical consistency enforced in [7]. In our work, we adopt the end-to-end solution for a regression problem instead of a classification one, and we also introduce a post-filtering stage, which consists of a Gaussian filtering that smoothie the resulting saliency map.

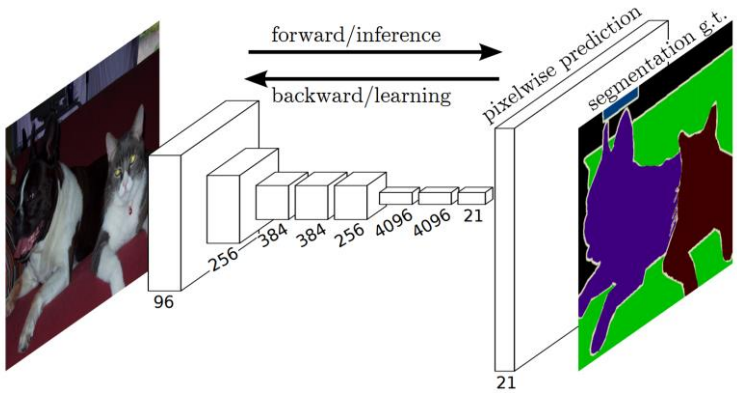

Figure 5 Model pipeline of DeepLab [5]

3. Methodology / project development:

This section presents how we solve the problem of saliency prediction, using a convent trained from scratch. The parameters of our network are learned by minimizing an Euclidean loss function defined directly on the ground truth saliency maps.

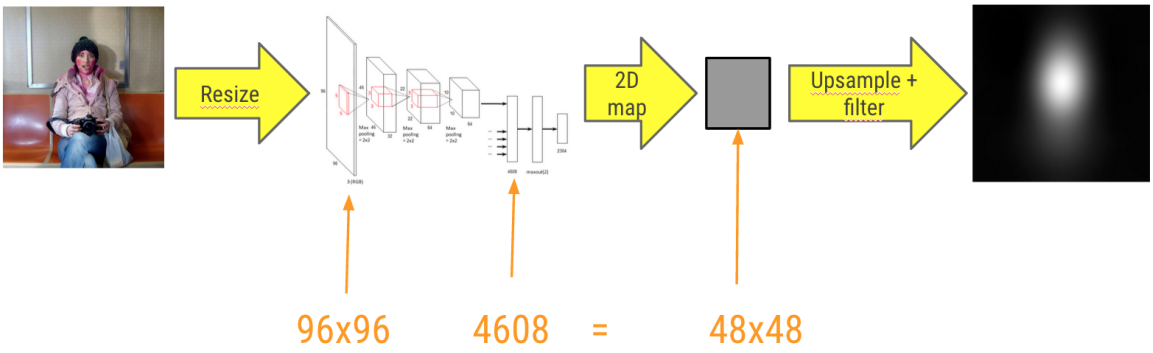

Figure 6 Pipeline of our ConvNet model.

3.1. Convolutional Neural Network

Neural Networks are machine learning training algorithms which exploit multiple layers of non-linear information processing for pattern analysis and classification, being inspired by how the human brain works. Neurons are the basic computational unit, each neuron performs a dot product with the input and its weights, then it adds the bias and applies the activation function (non-linearity).

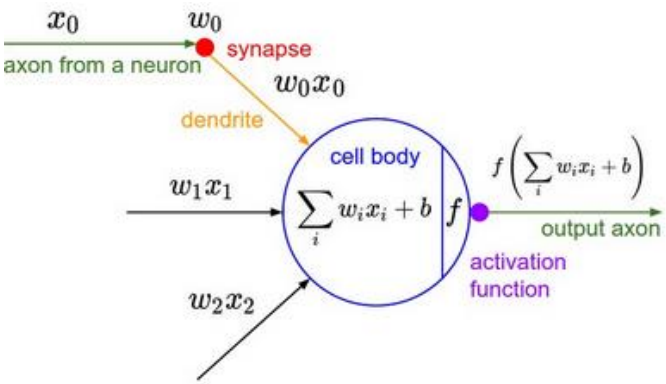

Figure 7 Mathematical model of a neuron inside the Neural Network.

Neural Networks are modelled as layers (grouped neurons) that are connected in acyclic graph. This layered architecture enables very efficient computation based on matrix multiplications interwoven with the application of the non-linearity.

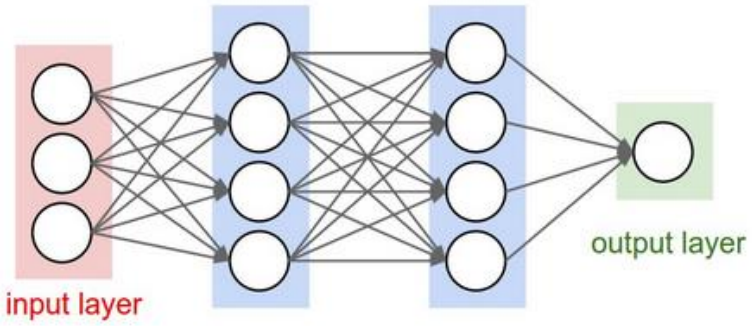

Figure 8 A 3-layer Neural Network with 3 inputs and 1 output

Once the architecture is chosen, in order to train Neural Networks, back propagation is applied to compute the gradients on the connections of the networks, with respect to a loss function.

Convolutional Neural Networks (ConvNets) are a class of Neural Networks which are made to process images as input data. The layers of a ConvNet have neuron organized in 3 dimensions: width, height, depth. So that each layer accepts 3D input and transforms it to a 3D output. Due to the overftting problem caused by the millions of parameters of the ConvNet, it usually use the same weight vector for each single depth slice, then the forward computation of the layers in each depth slice is performed as a convolution of neuron's weights and the input.

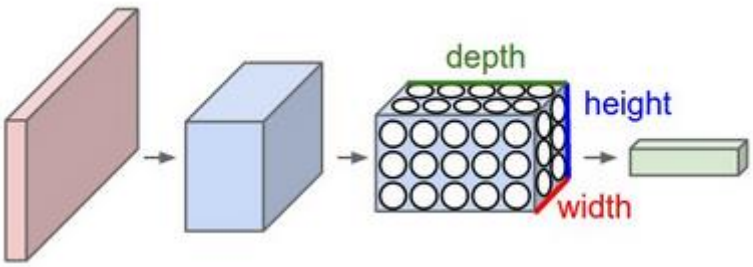

Figure 9 A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations.

.

There are four main types of layers to build ConvNet architectures:

Convolutional Layer: Is the core building block of the network. Its parameters consist of a set of learnable fileters. A dot product is compute between the filters and the input, the ConvNet will learn filters that activate when some specific type of feature in the input is detected.

Pooling Layer: It reduces the spatial size of the input in order to diminish the number of parameters and computation in the ConvNet.

Fully Connected Layer: Neurons between two adjacent layers are fully pair wise connected, while in neurons from the same layer are not. They can be interpreted as 1x1 convolutional layers.

ReLU Layer (non-linearity): The Rectified Linear Unit computes the function of $f(x) = \max(0, x)$ which is a threshold at zero.

3.2. Architecture

The detailed architecture of our network is illustrated in Figure 10. The network contains five learned layers: three convolutional layers and two fully connected layers.

The proposed architecture is not very deep if compared to other networks in the state of the art. Popular architectures trained on the 1, 200, 000 images of the ILSRVC 2012 challenge proposed from 7 [14] to 22 layers [18]. Our network is defined by only 5 layers which are trained separately on two training datasets collections of diverse sizes: 6, 000 for iSUN and 10, 000 for SALICON. This adopted shallow depth tries to prevent the overfitting problem, which is a great risk for models with a large amount of parameters, such as ConvNet.

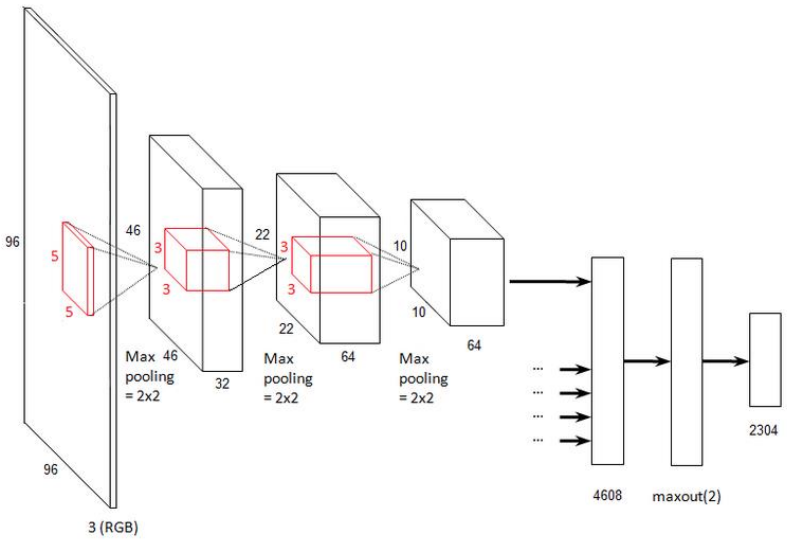

Figure 10 ConvNet architecture of our network.

The detailed description of the ConvNet stages is the following:

1. The input volume has size of [96x96x3] (RGB image), a size smaller than the [227x227x3] proposed in AlexNet. Similarly to the shallow depth, this design parameter is motivated by the smaller size of the training dataset if compared with the 1.2M images from ImageNet ILSRVC 2012.

2. The receptive field of the first 2D convolution is of size [5x5], whose outputs define a convolutional layer with 32 neurons. This layer is followed by a ReLU activation layer which applies an element wise non-linearity. Later, a max pooling layer progressively reduces the spatial size of the input image. Despite the loss of visual resolution at the output, this reduction also reduces the amount of model parameters and prevents overfitting. The max-pooling layer selects the maximum value of every [2x2] region, taking strides of two pixels.

3. The output of the previous stage has a size of [46x46x32]. The receptive field of this second stage is [3x3]. Again, this is followed by a ReLU layer and a max-pooling layer of size [2x2].

4. Finally, the last convolutional layer is fed with an input of size [22x22x64]. The receptive of this layer is also of [3x3] and it has 64 neurons. A ReLU and max pooling layers are stacked too.

5. A first fully connected layer receives the output of the third convolutional layer with a dimension of [10x10x64]. It contains a total of 4,608 neurons.

6. The second fully connected layer consists of a maxout layer with 2,304 neurons. The maxout operation [8] computes the pairs of the previous layers output.

7. Finally, the output of the last maxout layer is the saliency prediction array. The array is reshaped to have 2D dimensions and resized to the stimuli image size. Finally, a 2D Gaussian filter of a standard deviation of three is applied.

3.3. Training parameters

The limited amount of training data for our architecture made overfitting a significant challenge, so we used different techniques to minimize its effects.

Firstly, we apply norm constraint regularization for the maxout layers [8]. Secondly, we use data augmentation technique by mirroring all images. We also tested a dropout layer [10] after the first fully connected layer, with a dropout ratio of 0.5 (50% of probability to set a neurons output value to zero). However, this did not make much of a difference, so it is not included to the final model.

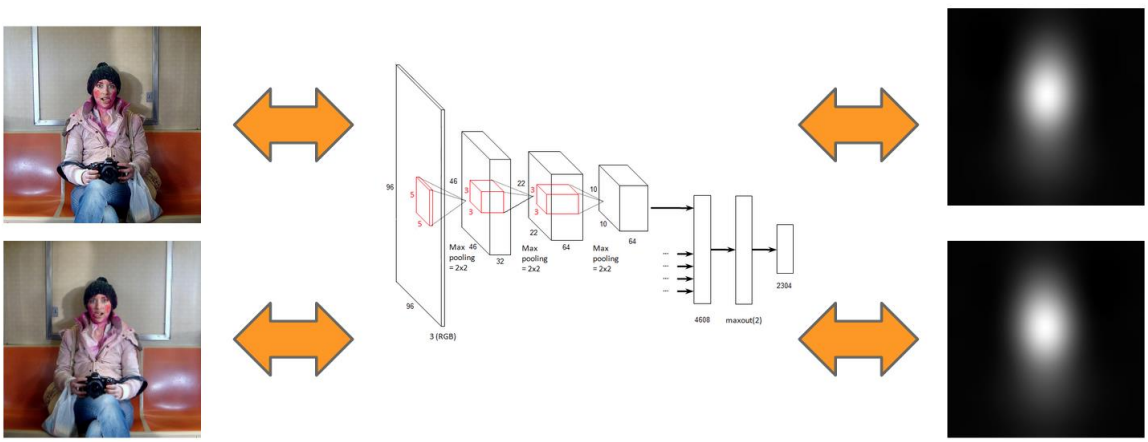

Figure 11 Mirror the training dataset.

3.3.1. Initialization of layer weights

The weights in all layers are initialized from a normal Gaussian distribution with zero mean and a standard deviation of 0.01, with biases initialized to 0.1.

Ground truth values that we used for training are saliency maps with normalized values between 0 and 1.

3.3.2. Dataset partitions

For validation control purposes, we split the training partitions of iSUN and SALICON datasets into 80% for training and the rest for real time validation.

The network was trained with stochastic gradient descent (SGD) and Nesterov momentum SGD optimization method that helps the loss function to converge faster.

The learning rate was changing over time; it started with a higher learning rate 0.03 and decreased during the course of training until 0.0001.

We set 1,000 epochs to train a separate network for each dataset.

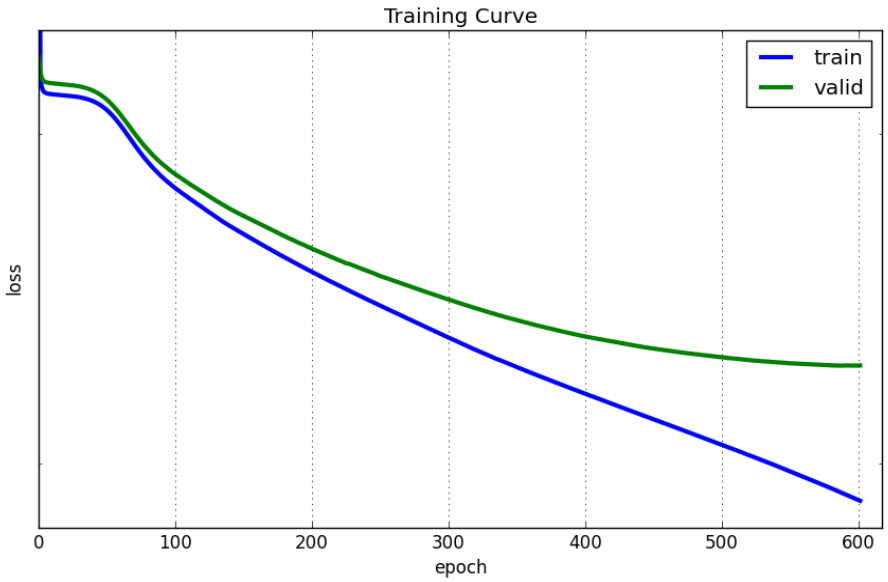

Figure 12 Learning curves for iSUN models.

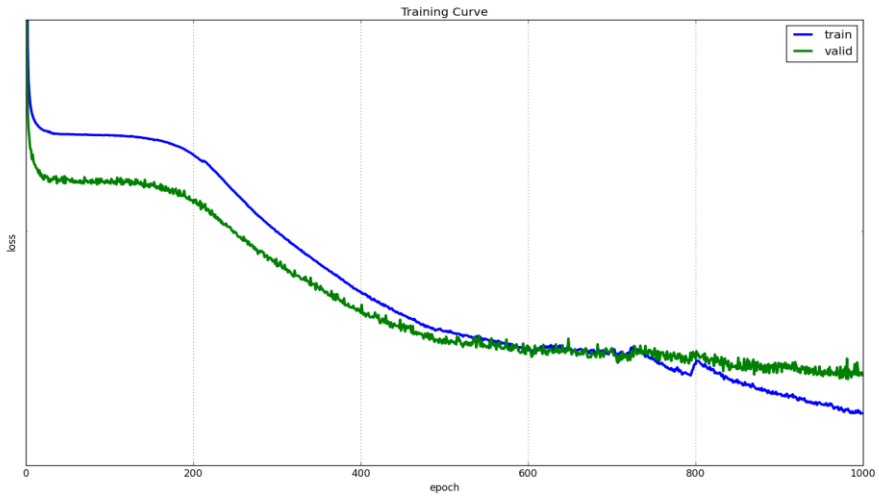

Figure 13 Learning curves for SALICON models.

Figures 12 and 13 present the learning curves for the iSUN and SALICON models, respectively. Fixations datasets are relatively limited in size, because is tedious and time consuming to collect. There are no more than 10.000 in each datasets comparing to the 1.2 million images in ILSVRC 2014. Typically to train a ConvNet, it needs massive amounts of data for training.

4. Results

4.1. LSUN Saliency Prediction Challenge 2015

The network was tested in the two datasets proposed in the LSUN challenge [23]:

iSUN [21]: a ground truth of gaze traces on images from the SUN dataset [20]. The collection is partitioned into 6,000 images for training, 926 for validation and 2,000 for test.

SALICON [12]: cursor clicks on the objects of interest from images of the Microsoft COCO dataset [16]. The collection contains 10,000 training images, 5,000 for validation and 5,000 for test.

Our solution is implemented using Python, NumPy and the deep learning library Theano [3, 2]. Processing was performed on an NVidia GPU GTX 980 with 2048 CUDA cores and 4GB of RAM. Our network took between six to seven hours to train for the SALICON dataset, and five to six hours for the iSUN dataset. Every saliency prediction requires 200ms per image.

We assessed our model on the LSUN saliency prediction challenge 2015 [23]. Table 1 and Table 2 present our results for iSUN and SALICON datasets. The model was evaluated separately on the test data of each datasets. The evaluation metrics were adopted from the MIT saliency benchmark [13, 4]. Our network consistently won the first place of the LSUN challenge in all metrics considered. Some qualitative results are also provided in Figure 14.

Method	Similarity	CC	AUC_shuffled	AUC_Borji	AUC_Judd
Our work	0.6833	0.8230	0.6650	0.8463	0.8693
Xidian	0.5713	0.6167	0.6484	0.7949	0.8207
WHU_IIP	0.5593	0.6263	0.6307	0.7960	0.8197
LCYLab	0.5474	0.5699	0.6259	0.7921	0.8133
Rare 2012 Improved	0.5199	0.5199	0.6283	0.7582	0.7846
Baseline: BMS[22]	0.5026	0.3465	0.5885	0.6560	0.6914
Baseline: GBVS[24]	0.4798	0.5087	0.6208	0.7913	0.8115
Baseline: Itti[11]	0.4251	0.3728	0.6024	0.7262	0.7489

Table 1 Results of the LSUN challenge 2015 for saliency prediction with the iSUN dataset.

Method	Similarity	CC	AUC_shuffled	AUC_Borji	AUC_Judd
Our work	0.5198	0.5957	0.6698	0.8291	0.8364
WHU_IIP	0.4908	0.4569	0.6064	0.7759	0.7923
Rare 2012 Improved	0.5017	0.5108	0.6644	0.8047	0.8148
Xidian	0.4617	0.4811	0.6809	0.7990	0.8051
Baseline: BMS[22]	0.4542	0.4268	0.6935	0.7699	0.7899
Baseline: GBVS[24]	0.4460	0.4212	0.6303	0.7816	0.7899
Baseline: Itti[11]	0.3777	0.2046	0.6101	0.6603	0.6669

Table 2 Results of the LSUN challenge 2015 for saliency prediction with the SALICON dataset.

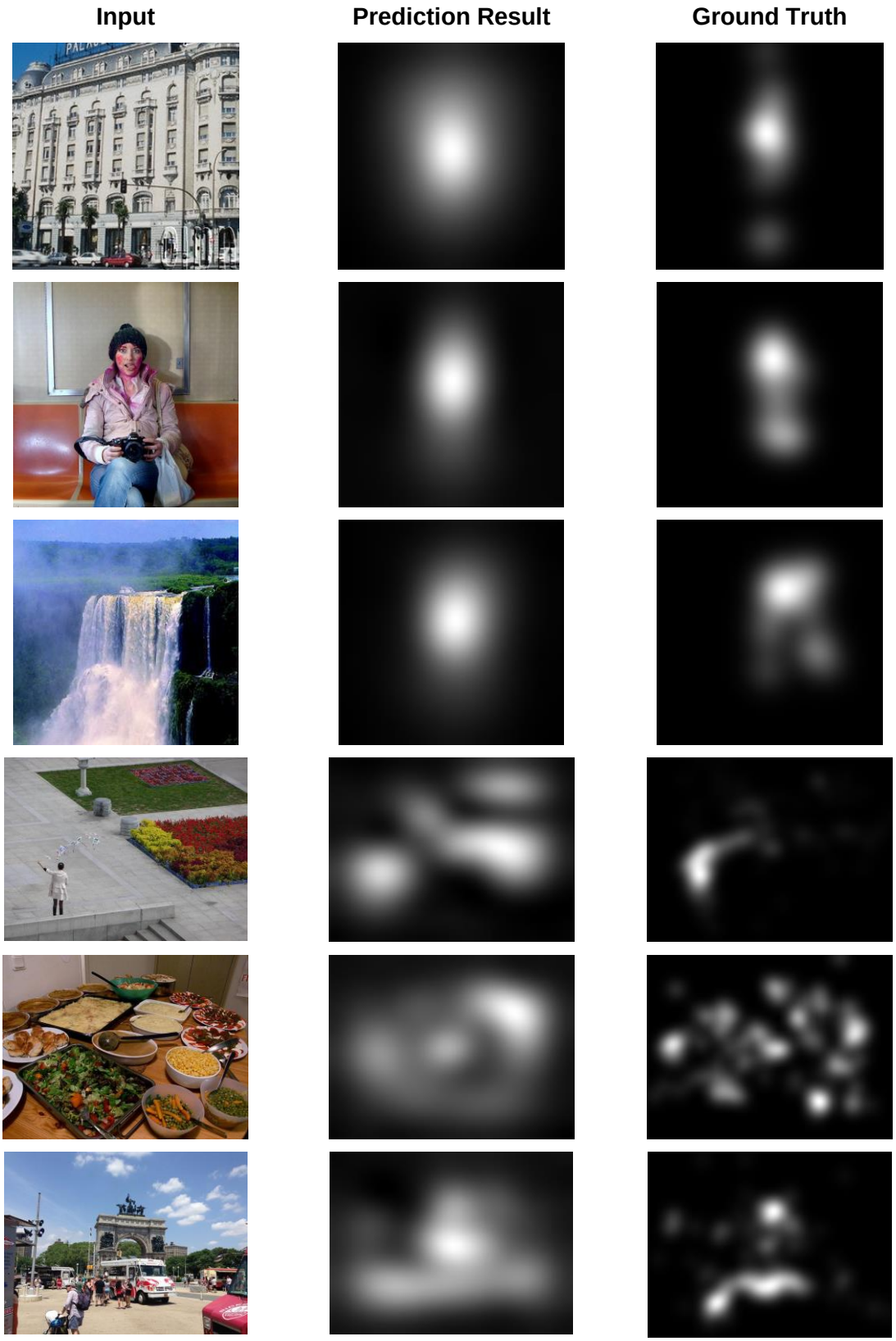

Figure 14 Saliency maps generated by JuntingNet. The first column corresponds to the input image, the second column the prediction from JuntingNet and the third one and the third on to the provided ground truth. First three rows correspond to images from the iSUN

4.2. MIT 300 Benchmark

In the latest stage of this thesis, we evaluated our iSUN model on the MIT benchmark dataset [26] containing 300 natural images with eye tracking data from 39 observers. In this case results were much more modest, in a position that would rank #23 if published at the time of writing this report. We hypothesize that this average performance is due to the fact that we never fine-tuned our network with the data used in the MIT 300, so our results may be affected by a dataset bias [25]. In addition, our end-to-end solution did not introduce any posterior bias towards the center of the image, a common practice in many other solutions which, by itself, outperforms our technique. **Error! Reference source not ound.** presents our results, compared with a few selected examples. The full list of submissions can be accessed online [1]

Method	Similarity	CC	AUC_shuffled	AUC_Borji	AUC_Judd
Baseline: infinite humans	1	1	0.80	0.87	0.91
Deep Gaze 1 [15]	0.39	0.48	0.66	0.85	0.84
eDN [19]	0.41	0.45	0.62	0.81	0.82
Our work	0.4708	0.4285	0.5075	0.7416	0.7720
Baseline: Center	0.39	0.38	0.51	0.77	0.78

Table 3 Selected results on MIT 300 benchmark.

[1] http://saliency.mit.edu/results_mit300.html

5. Budget

This research project has been developed using software libraries and frameworks that are free for both academic and commercial use, so the cost of the project comes mainly from the time spent by the researchers who have involved in it:

	Amount	Wage	Hours spent	Total
Junior engineer	1	8.00€/h	400h	3,200€
Senior engineer	1	20.00€/h	80h	1,600€

TOTAL: **4,800€**

6. Conclusions and future development:

This work has presented an end-to-end ConvNet for saliency prediction, trained only with datasets of visual saliency data provided in the LSUN challenge. Up to the author's knowledge, this was the first ConvNet to be trained end-to-end at the time of its publication.

JuntingNet won the LSUN saliency prediction challenge 2015 with a large margin with respect to other participants, with an award of $300 donated by Amazon A9. The award diploma is included in the annexes of this thesis report. The prize was also highlighted on the front page of our ETSETB Telecom School[2].

Our results demonstrate that a not very deep ConvNet is capable of achieving impressive results on a highly challenging task, especially because we are adjusting the amount of parameters of the model to the amount of available data. It must be highlighted that no additional data was used during the estimation of JuntingNet.

All of our experiments suggest that our results can be improved with larger datasets and longer training time. We think that increasing the amount of fixation data and longer training time we can achieve a better result. An obvious next step suggested by this to convert the network into a fully convolutional ConvNet like the DeepLab model for segmentation. Furthermore, it is also interesting to visualize the learned filters to evaluate the possibility to add more layers in order to have a deeper architecture.

We hope that with our success of using deep learning techniques in the saliency prediction task can awake more interest between researchers to apply this technique on other more complex computer vision problems.

This work was also shared with the scientific community through a preprint paper in arXiv [27] and the publication of the source code, trained models and additional examples from http://bit.ly/juntingnet.

We expect to refine our model and results on the MIT 300 dataset for a future submission in the regular International Conference on Computer Vision and Pattern Recognition 2016, whose deadline is in the coming October 2015.

[2] http://www.etsetb.upc.es/mason-share/notif/2806.html

Bibliography:

[1] P. Agrawal, D. Stansbury, J. Malik, and J. L. Gallant. Pixels to voxels: Modeling visual representation in the human brain. arXiv preprint arXiv:1407.5104, 2014.
[2] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. arXiv preprint arXiv:1211.5590, 2012.
[3] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a cpu and gpu math expression compiler. In Proceedings of the Python for scientific computing conference (SciPy), volume 4, page 3. Austin, TX, 2010.
[4] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. Mit saliency benchmark. http://saliency.mit.edu/.
[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. CoRR, abs/1412.7062, 2014.
[6] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, 2014.
[7] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8):1915– 1929, 2013.
[8] [8] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. arXiv preprint arXiv:1302.4389, 2013.
[9] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In Advances in neural information processing systems, pages 545–552, 2006.
[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.
[11] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis & Machine Intelligence, (11):1254–1259, 1998.
[12] M. Jiang, S. Huang, J. Duan, and Q. Zhao. SALICON: Saliency in context. In Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015.
[13] T. Judd, F. Durand, and A. Torralba. A benchmark of computational models of saliency to predict human fixations. In MIT Technical Report, 2012.
[14] 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.
[15] M. Kummerer, L. Theis, and M. Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. arXiv preprint arXiv:1411.1045, 2014.
[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014, pages 740–755. Springer, 2014.
[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014.
[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. arXiv preprint arXiv:1409.4842, 2014.
[19] E. Vig, M. Dorr, and D. Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2798–2805. IEEE, 2014.
[20] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, et al. Sun database: Large-scale scene recognition from abbey to zoo. In Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, pages 3485–3492. IEEE, 2010.
[21] 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. pi
[22] J. Zhang and S. Sclaroff. Saliency detection: A boolean map approach. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 153–160. IEEE, 2013.
[23] Y. Zhang, F. Yu, S. Song, P. Xu, A. Seff, and J. Xiao. Largescale scene understanding challenge: Eye tracking saliency estimation.
[24] Harel, Jonathan, C. Koch, and P. Perona. "Graph-based visual saliency." Advances in neural information processing systems. 2006.
[25] Torralba, Antonio, and Alexei Efros. "Unbiased look at dataset bias." *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011.
[26] Judd, Tilke, Frédo Durand, and Antonio Torralba. "A benchmark of computational models of saliency to predict human fixations." (2012).
[27] Junting Pan and Xavier Giró-i-Nieto. End-to-end convolutional network for saliency prediction. arXiv preprint arXiv:1507.01422, 2015

Appendices

Diploma of the LSUN Challenge

Glossary and vocabulary

ConvNet: Convolutional Neural Network

Epoch: Consist in the time that the neural network to train witch every training samples on in one pass.

CNN: Convolutional Neural Network

LSUN: Large-scale Scene Understanding