



Universitat Autònoma de Barcelona

ADVERTIMENT. L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  http://cat.creativecommons.org/?page_id=184

ADVERTENCIA. El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

WARNING. The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>



**Universitat Autònoma
de Barcelona**

Visual Saliency for Object Recognition,
and Object Recognition for Visual Saliency

A dissertation submitted by **Carola Figueroa Flores**
at Universitat Autònoma de Barcelona to fulfil the
degree of **Doctor of Philosophy**.

Bellaterra, February 2, 2021

Directors	Dr. Joost van de Weijer Dr. Bogdan Raducanu Centre de Visió per Computador
Thesis committee	Dr. Javier Vazquez Corral Dpto. Tecnologies de la Informació i les Comunicacions Universidad Pompeu Fabra Dr. Alejandro Parraga Dpto. Ciències de la Computació and Centre de Visió per Computador Universitat Autònoma de Barcelona Dr. Xosé Manuel Pardo López Dpto. Electrónica e Computación Universidade de Santiago de Compostela / CiTIUS
Supplent	Dr. Xavier Otazu Dpto. Ciències de la Computació and Centre de Visió per Computador Universitat Autònoma de Barcelona Dr. Adrià Ruiz Institut de Robòtica i Informàtica industrial (CSIC-UPC)



This document was typeset by the author using $\text{\LaTeX} 2_{\epsilon}$.

The research described in this book was carried out at the Centre de Visió per Computador, Universitat Autònoma de Barcelona. Copyright © 2021 by **Carola Figueroa Flores**. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author.

ISBN: 978-84-948531-9-7

Printed by Ediciones Gráficas Rey, S.L.

“Learning is a Lifelong Process”

We now accept the fact that learning is a lifelong process of keeping abreast of
change.

And the most pressing task is to teach people how to learn

— Peter Drucker

Dedicated to my parents, Mirta and Patricio .

Acknowledgements

First of all, I would like to thank my tutors Joost and Bogdan, who during this program have been the key to becoming a competent researcher. They, in addition to supporting me in the scientific field, became friends to support me when I faced new challenges. Thank you for being patient with me when I was a confused student trying to understand some definitions or when I had no progress in my experiments. Although I think the latter was a headache for them, they didn't give up on me.

I have been fortunate to meet many inspiring people at CVC. In particular, I would like to thank: Abel González and David Berga for their valuable comments on my research, since thanks to their help I have been able to publish and participate in conferences.

I want to thank many people who made my stay in Spain pleasant and comfortable. Thanks to all the CVC staff, a lot of my time was spent there and it was like living at home. In addition, I want to thank my friends and fellow PhD students who made my stay at the UAB an incredible experience: Pau Riba, Arnau Barro, Xavier Soria, Pau Rodríguez, Albert Berengel, Asma Bensalah, Marc Masana and many others. And to my dear friends Claudia Alexandra, Claudia Bullion, Alicia (Xiaoxi Wu), my flatmates (Laura, Julia and Maria) and Lorena Gimenez, to whom I will always be grateful for making me feel so welcomed and welcomed in her family.

I am especially grateful to CONICYT, an institution of the Chilean government that supported me with the scholarship for the PhD program. Also, many thanks to my University of Bío-Bío, which gave me the opportunity to specialize in the area of computer vision, and to my colleagues from the Department of Computer Science and Information Technology, who supported and encouraged me in every moment.

Acknowledgements

Last but not least, I would like to show my gratitude to my family, including my grandmother, my parents, my aunt Lucy and my brothers, for their continuous support throughout my whole life. I am grateful to my boyfriend Gonzalo for his unconditional trust and support, especially during the most difficult period of my study (this pandemic).

Abstract

For humans, the recognition of objects is an almost instantaneous, precise and extremely adaptable process. Furthermore, we have the innate capability to learn new object classes from only few examples. The human brain lowers the complexity of the incoming data by filtering out part of the information and only processing those things that capture our attention. This, mixed with our biological predisposition to respond to certain shapes or colors, allows us to recognize in a simple glance the most important or salient regions from an image. This mechanism can be observed by analyzing on which parts of images subjects place attention; where they fix their eyes when an image is shown to them. The most accurate way to record this behavior is to track eye movements while displaying images.

Computational saliency estimation aims to identify to what extent regions or objects stand out with respect to their surroundings to human observers. Saliency maps can be used in a wide range of applications including object detection, image and video compression, and visual tracking. The majority of research in the field has focused on automatically estimating saliency maps given an input image. Instead, in this thesis, we set out to incorporate saliency maps in an object recognition pipeline: we want to investigate whether saliency maps can improve object recognition results.

In this thesis, we identify several problems related to visual saliency estimation. First, to what extent the estimation of saliency can be exploited to improve the training of an object recognition model when scarce training data is available. To solve this problem, we design an image classification network that incorporates saliency information as input. This network processes the saliency map through a dedicated network branch and uses the resulting characteristics to modulate the standard bottom-up visual characteristics of the original image input. We will refer

Acknowledgements

to this technique as saliency-modulated image classification (SMIC). In extensive experiments on standard benchmark datasets for fine-grained object recognition, we show that our proposed architecture can significantly improve performance, especially on dataset with scarce training data.

Next, we address the main drawback of the above pipeline: SMIC requires an explicit saliency algorithm that must be trained on a saliency dataset. To solve this, we implement a hallucination mechanism that allows us to incorporate the saliency estimation branch in an end-to-end trained neural network architecture that only needs the RGB image as an input. A side-effect of this architecture is the estimation of saliency maps. In experiments, we show that this architecture can obtain similar results on object recognition as SMIC but without the requirement of ground truth saliency maps to train the system.

Finally, we evaluated the accuracy of the saliency maps that occur as a side-effect of object recognition. For this purpose, we use a set of benchmark datasets for saliency evaluation based on eye-tracking experiments. Surprisingly, the estimated saliency maps are very similar to the maps that are computed from human eye-tracking experiments. Our results show that these saliency maps can obtain competitive results on benchmark saliency maps. On one synthetic saliency dataset this method even obtains the state-of-the-art without the need of ever having seen an actual saliency image for training.

Key words: computer vision, visual saliency, fine-grained object recognition, convolutional neural networks, images classification.

Resumen

El reconocimiento de objetos para los seres humanos es un proceso instantáneo, preciso y extremadamente adaptable. Además, tenemos la capacidad innata de aprender nuevas categorías de objetos a partir solamente de unos pocos ejemplos. El cerebro humano reduce la complejidad de los datos entrantes filtrando parte de la información y procesando las cosas que captan nuestra atención. Esto, combinado con nuestra predisposición biológica a responder a determinadas formas o colores, nos permite reconocer en una simple mirada las regiones más importantes o destacadas de una imagen. Este mecanismo se puede observar analizando en qué partes de las imágenes los sujetos ponen su atención; por ejemplo donde fijan sus ojos cuando se les muestra una imagen. La forma más precisa de registrar este comportamiento es rastrear los movimientos de los ojos mientras se muestran imágenes.

La estimación computacional del “saliency”, tiene como objetivo diseñar algoritmos que, dada una imagen de entrada, estimen mapas de “saliency”. Estos mapas se pueden utilizar en una variada gama de aplicaciones, incluida la detección de objetos, la compresión de imágenes y videos y el seguimiento visual. La mayoría de la investigación en este campo se ha centrado en estimar automáticamente estos mapas de “saliency”, dada una imagen de entrada. En cambio, en esta tesis, nos propusimos incorporar la estimación de “saliency” en un procedimiento de reconocimiento de objeto, puesto que, queremos investigar si los mapas de “saliency” pueden mejorar los resultados de la tarea de reconocimiento de objetos.

En esta tesis, identificamos varios problemas relacionados con la estimación del “saliency” visual. Primero, pudimos determinar en qué medida se puede aprovechar la estimación del “saliency” para mejorar el entrenamiento de un modelo de reconocimiento de objetos cuando se cuenta con escasos datos de entrenamiento.

Acknowledgements

Para resolver este problema, diseñamos una red de clasificación de imágenes que incorpora información de "saliency" como entrada. Esta red procesa el mapa de "saliency" a través de una rama de red dedicada y utiliza las características resultantes para modular las características visuales estándar ascendentes de la entrada de la imagen original. Nos referiremos a esta técnica como clasificación de imágenes moduladas por prominencia (SMIC en inglés). En numerosos experimentos realizando sobre en conjuntos de datos de referencia estándar para el reconocimiento de objetos "fine-grained", mostramos que nuestra arquitectura propuesta puede mejorar significativamente el rendimiento, especialmente en conjuntos de datos con datos con escasos datos de entrenamiento. Luego, abordamos el principal inconveniente del problema anterior: es decir, SMIC requiere explícitamente un algoritmo de "saliency", el cual debe entrenarse en un conjunto de datos de "saliency". Para resolver esto, implementamos un mecanismo de alucinación que nos permite incorporar la rama de estimación de "saliency" en una arquitectura de red neuronal entrenada de extremo a extremo que solo necesita la imagen RGB como entrada. Un efecto secundario de esta arquitectura es la estimación de mapas de "saliency". En variados experimentos, demostramos que esta arquitectura puede obtener resultados similares en el reconocimiento de objetos como SMIC pero sin el requisito de mapas de "saliency" para entrenar el sistema. Finalmente, evaluamos la precisión de los mapas de "saliency" que ocurren como efecto secundario del reconocimiento de objetos. Para ello, utilizamos un de conjuntos de datos de referencia para la evaluación de la prominencia basada en experimentos de seguimiento ocular. Sorprendentemente, los mapas de "saliency" estimados son muy similares a los mapas que se calculan a partir de experimentos de seguimiento ocular humano. Nuestros resultados muestran que estos mapas de "saliency" pueden obtener resultados competitivos en mapas de "saliency" de referencia. En un conjunto de datos de "saliency" sintético, este método incluso obtiene el estado del arte de la técnica, sin la necesidad de haber visto nunca una imagen de "saliency" real para el entrenamiento.

Palabras claves: visión por computadora, "saliency" visual, reconocimiento de objetos "fine-grained", redes neuronales convolucionales, clasificación de imágenes.

Resum

Per als humans, el reconeixement d'objectes és un procés gairebé instantani, precís i extremadament adaptable. A més, tenim la capacitat innata d'aprendre classes d'objectes nous a partir d'uns pocs exemples. El cervell humà redueix la complexitat de les dades entrants filtrant part de la informació i processant només aquelles coses que ens capturen l'atenció. Això, barrejat amb la nostra predisposició biològica per respondre a determinades formes o colors, ens permet reconèixer en un simple cop d'ull les regions més importants o destacades d'una imatge. Aquest mecanisme es pot observar analitzant sobre quines parts de les imatges hi posa l'atenció; on es fixen els ulls quan se'ls mostra una imatge. La forma més precisa de registrar aquest comportament és fer un seguiment dels moviments oculars mentre es mostren imatges.

L'estimació computacional de la salubritat té com a objectiu identificar fins a quin punt les regions o els objectes destaquen respecte als seus entorns per als observadors humans. Els mapes Saliency es poden utilitzar en una àmplia gamma d'aplicacions, inclosa la detecció d'objectes, la compressió d'imatges i vídeos i el seguiment visual. La majoria de les investigacions en aquest camp s'han centrat en estimar automàticament els mapes de salubritat donats una imatge d'entrada. En el seu lloc, en aquesta tesi, ens proposem incorporar mapes de salubritat en una canalització de reconeixement d'objectes: volem investigar si els mapes de salubritat poden millorar els resultats del reconeixement d'objectes.

En aquesta tesi, identifiquem diversos problemes relacionats amb l'estimació de la salubritat visual. En primer lloc, fins a quin punt es pot aprofitar l'estimació de la salubritat per millorar la formació d'un model de reconeixement d'objectes quan es disposa de dades d'entrenament escasses. Per solucionar aquest problema, dissenyem una xarxa de classificació d'imatges que incorpori informació d'informació

Acknowledgements

salarial com a entrada. Aquesta xarxa processa el mapa de saliència a través d'una branca de xarxa dedicada i utilitza les característiques resultants per modular les característiques visuals estàndard de baix a dalt de l'entrada d'imatge original. Ens referirem a aquesta tècnica com a classificació d'imatges modulades en salinitat (SMIC). En amplis experiments sobre conjunts de dades de referència estàndard per al reconeixement d'objectes de gra fi, demostrem que la nostra arquitectura proposada pot millorar significativament el rendiment, especialment en el conjunt de dades amb dades de formació escasses.

A continuació, abordem l'inconvenient principal de la canonada anterior: SMIC requereix un algorisme de saliència explícit que s'ha de formar en un conjunt de dades de saliència. Per solucionar-ho, implementem un mecanisme d'al·luciniació que ens permet incorporar la branca d'estimació de la salubritat en una arquitectura de xarxa neuronal entrenada de punta a punta que només necessita la imatge RGB com a entrada. Un efecte secundari d'aquesta arquitectura és l'estimació de mapes de salubritat. En experiments, demostrem que aquesta arquitectura pot obtenir resultats similars en reconeixement d'objectes com SMIC, però sense el requisit de mapes de salubritat de la veritat del terreny per entrenar el sistema.

Finalment, hem avaluat la precisió dels mapes de salubritat que es produeixen com a efecte secundari del reconeixement d'objectes. Amb aquest propòsit, fem servir un conjunt de conjunts de dades de referència per a l'avaluació de la validesa basats en experiments de seguiment dels ulls. Sorprenentment, els mapes de salubritat estimats són molt similars als mapes que es calculen a partir d'experiments de rastreig d'ulls humans. Els nostres resultats mostren que aquests mapes de salubritat poden obtenir resultats competitius en els mapes de salubritat de referència. En un conjunt de dades de saliència sintètica, aquest mètode fins i tot obté l'estat de l'art sense la necessitat d'haver vist mai una imatge de saliència real.

Paraules clau: visió per ordinador, saliència visual, reconeixement d'objectes de gra fi, xarxes neuronals convolucionals, classificació d'imatges.



Contents

Acknowledgements	i
Abstract (English/Spanish)	iii
List of figures	xiii
List of tables	xvii
1 Introduction	1
1.1 Challenges in Saliency for Object Recognition	2
1.2 Research Objectives	5
1.3 Contributions and Outline	6
2 Related Work	9
2.1 Learned Representations	9
2.1.1 Convolutional Neural Networks	12
2.1.2 Neural Networks for Scarce Data Domains	13
2.2 Visual Saliency	15
2.2.1 Saliency Methods	17
	ix

Contents

2.2.2	Saliency for Image Classification	23
2.2.3	Link between attention and saliency	23
2.2.4	Center bias	24
2.2.5	Datasets for Saliency Estimation	26
2.3	Fine-grained Object Recognition	28
3	Saliency for Fine-grained Object Recognition	31
3.1	Introduction	31
3.2	Saliency Modulation for Scarce Data Object Classification	33
3.2.1	Combining RGB with Saliency for Image Classification	35
3.2.2	Training the Saliency Branch	36
3.2.3	Saliency input	37
3.3	Experiments	39
3.3.1	Experimental Setup	39
3.3.2	Experimental Results	40
3.4	Conclusion	51
4	Hallucinating Saliency Maps for Fine-Grained Image Classification for Limited Data Domains	53
4.1	Introduction	53
4.2	Proposed Method	55
4.2.1	Overview of the Method	55
4.2.2	Hallucination of saliency maps from RGB images	56
4.2.3	Fusion of RGB and Saliency Branches	57
4.2.4	Training on Imagenet and fine-tuning on a target dataset	58

4.3	Experimental Results	59
4.3.1	System setup	59
4.3.2	Fine-grained Image Classification Results	60
4.4	Conclusion	62
5	Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition	69
5.1	Introduction	69
5.2	Proposed Approach	71
5.2.1	Network architecture	71
5.2.2	Training the saliency branch	74
5.2.3	Combination with center bias	74
5.3	Experimental Results	76
5.3.1	Setup	76
5.3.2	Results	78
5.4	Conclusion	82
6	Conclusions and Future Work	93
6.1	Conclusions	93
6.1.1	List of Publications	94
6.2	Future Work	94
	Bibliography	118

List of Figures

1.1	Example of saliency map based on eye fixations.	3
2.1	Illustration of a deep learning model from [57].	11
2.2	Convolutional Neural Network for image processing, e.g., handwriting recognition [99].	12
2.3	Illustration of the architecture of the AlexNet CNN introduced in [1].	13
2.4	The VGG16 architecture [113].	14
2.5	Residual block with skip connection [62]	15
2.6	Example of salient object in natural image. a) original image; b) ground truth ; c) example saliency detection results.	16
2.7	Saliency detection methods ontology [15]	17
2.8	An example of Attention map and Saliency map from an real world scene. [65].	25
3.1	Overview of our fine-grained recognition model using saliency information. We process the two inputs, RGB and Saliency map, through two convolutional layers and then fuse the resulting features with a modulation layer. We then continue processing the fused features with three more convolutional layers and three fully connected layers, ending with the final classification layer.	34

List of Figures

3.2	Saliency images generated with the different saliency estimation approaches considered, as well as the two baseline saliency maps evaluated, White and Center. We also include the original RGB image for reference.	38
3.3	Experiments on four datasets using iSEEL [174] to generate the saliency maps. <i>Baseline-RGB</i> is compared against two different ways to initialize the saliency branch of our model: from scratch and pretrained on ImageNet [158].	43
3.4	Some success examples on Flowers [133]: when the prediction done by Baseline-RGB fails to infer the right label for some test images, but the prediction by our approach is correct. From left to right: input image, saliency images generated with iSEEL [174], example image of the class with which the input image was wrongly predicted.	44
3.5	Some failure examples on Flowers [133]: when the prediction done by our method fails to infer the right label for some test images, but the prediction by Baseline-RGB is correct. From left to right: input image, saliency images generated with iSEEL [174], example image of the class with which the input image was wrongly predicted.	45
3.6	Average percentage of the total backpropagated gradient energy per epoch that is inside the bird bounding box. The graph shows that for our approach significantly more backpropagated gradient is on the relevant image region (for both the version trained from scratch and the version with pretrained saliency branch).	46
3.7	Correlation between the performance of the saliency method in terms of NSS and the fine-grained recognition accuracy of our method using the corresponding saliency model. Results with AlexNet [93] on Flowers [133].	48
4.1	Overview of our method. We process an RGB input image through two branches: one branch extracts the RGB features and the other one is used to learn saliency maps. The resulting features are merged via a modulation layer, which continues with a few more convolutional layers and a classification layer. The network is trained in two steps. .	56

4.2 Graph shows the classification accuracy on Flowers for various number of layers in the saliency branch. Best results are obtained with four convolutional layers. Baseline refers to the method without saliency branch. 63

4.3 Graph shows the classification accuracy on Flowers. Various points for fusing the saliency and RGB branch are evaluated. Best results are obtained when fusion is placed before the pool-2 layer. Baseline refers to the method without saliency branch. 64

5.1 Overview of our method. We process an RGB input image through two branches: one branch extracts the RGB features and the other one is used to learn saliency maps. 72

5.2 Qualitative results for real images (Toronto dataset). Each image is represented in a different column and each model saliency map in each row. The ground truth density map of human fixations is represented in the 2nd row. 81

5.3 Qualitative results for synthetic images (SID4VAM dataset). Each image is represented in a different column and each model saliency map in each row. The ground truth density map of human fixations is represented in the 2nd row. 82

List of Tables

2.1	Benchmark datasets for Fine-grained classification from [196]	29
3.1	Results for the baseline model and different variations of our architecture incorporating saliency information. The results correspond to the classification accuracy on the Flowers dataset [133] with AlexNet [93]. Each column indicates the number of training images used, and the rightmost column shows the average	41
3.2	Results on Flowers [133] with AlexNet [93] using two (S2) or three (S3) convolutional layers for the saliency branch.	42
3.3	Results on Flowers [133] with AlexNet [93] when reducing the number of parameters of the saliency branch.	42
3.4	Comparison of different saliency methods regarding the effect on our model. The results correspond to the classification accuracy on the Flowers dataset [133] when using our full model with AlexNet [93] as base network. Each column indicates the number of training images used, and the rightmost column shows the average.	47
3.5	Results for the baseline model and different variations of our architecture incorporating saliency information in different blocks. The results correspond to the classification accuracy on the Flowers dataset [133] with ResNet-50 [62]. Each column indicates the number of training images used, and the rightmost column shows the average	49

List of Tables

3.6	Results for the baseline model and different variations of our architecture incorporating saliency information in different blocks. The results correspond to the classification accuracy on the Flowers dataset [133] with ResNet-152 [62]. Each column indicates the number of training images used, and the rightmost column shows the average	49
3.7	Results on Flowers [133] using ResNet-50 and Resnet-152 [62] as base networks and SALICON [80] as saliency method.	50
3.8	Comparison with state of the art methods for domain-specific fine-grained recognition using the standard data splits of Flowers [133], Birds [199] and Cars [90]. Our approach uses ResNet-152 [62] as base network and SALICON [80] saliency maps.	50
3.9	Results for few-shot classification on Flowers [133] when using our full model with AlexNet [93] as base network.	51
4.1	Results on Flowers, Birds and Cars (results are the average over three runs), using AlexNet as base network. For the baseline model and different variations of layers on saliency branch of our architecture for saliency detection . The results correspond to the classification accuracy. Each column indicates the number of training images used, and the rightmost column shows the average.	61
4.2	Results on Flowers, Birds and Cars (results are the average over three runs), using AlexNet as base network. For the baseline model and different variations of our architecture incorporating the merge of our hallucinating saliency. The results correspond to the classification accuracy. Each column indicates the number of training images used, and the rightmost column shows the average.	62
4.3	Classification accuracy for Flowers, Cars, and Birds dataset (results are the average over three runs), using AlexNet as base network. Results are provided for varying number of training images, from 1 until 30; K refers to using the number of training images used in the official dataset split. The rightmost column shows the average. The * indicates that the method requires an explicit saliency method. Our method (Approach B) obtains similar results as SMIC but without the need of a pretrained saliency network trained on a saliency dataset. .	65

4.4	Classification accuracy for Flowers, Cars, and Birds dataset (results are the average over three runs), using ResNet152 as base network. Results are provided for varying number of training images, from 1 until 30; K refers to using the number of training images used in the official dataset split. The rightmost column shows the average. The * indicates that the method requires an explicit saliency method. Our method (Approach B) obtains similar results as SMIC but without the need of a pretrained saliency network trained on a saliency dataset.	65
4.5	Comparison with state of the art methods for domain-specific fine-grained recognition using the standard data splits of Flowers, Birds and Cars.	66
4.6	Some success examples on Flowers: when the prediction done by Baseline-RGB fails to infer the right label for some test images, but the prediction by our approach is correct. Example image contains image of the wrongly predicted class.	67
4.7	Some failure examples on Flowers: when the prediction done by our method fails to infer the right label for some test images, but the prediction by Baseline-RGB is correct. Example image contains image of the wrongly predicted class.	68
4.8	Visualization of saliency maps for Flowers-102, Birds and Cars.	68
5.1	Simulating the Center Bias by parametrizing Gaussian	75
5.2	Description of saliency models	77
5.3	Characteristics of eye tracking datasets	77
5.4	Benchmark of our method with different networks (top-1 networks are underlined)	78
5.5	Analysis of normalization on MIT1003, Toronto, KTH, CAT2000 and SID4VAM, using AlexNet as base-network	83
5.6	Analysis of adquisition on MIT1003, Toronto, KTH, CAT2000 and SID4VAM, using AlexNet as base-network	84
5.7	Analysis of GVA gaussian on real images dataset : MIT1003, Toronto and KTH.	85

List of Tables

5.8	Analysis of GVA gaussian on synthetics images dataset on CAT2000 and SID4VAM, using AlexNet as base network.	86
5.9	Analysis of acquisition on MIT1003, Toronto, KTH, CAT2000 and SID4VAM, using AlexNet as base-network	87
5.10	Ablation of fusion and normalization on all saliency datasets. We show results for the AUC-Judd metric (top-1 fusion is bold)	88
5.11	Qualitative results using human center bias	89
5.12	Comparison our saliency output with on standard benchmark methods over synthetic image datasets (Left: Toronto, Right: SID4VAM) for saliency prediction. (Top) Baseline low-level saliency models. (Bottom) State-of-the-art deep saliency models. Best score for each metric is defined as bold and TOP-3 scores are <u>underlined</u>	90
5.13	Comparison our saliency output with on standard benchmark methods over synthetic image datasets (Left: Toronto, Right: SID4VAM) for saliency prediction. (Top) Baseline low-level saliency models. (Bottom) State-of-the-art deep saliency models. Best score for each metric is defined as bold and TOP-3 scores are <u>underlined</u>	90
5.14	Benchmark on saliency models. We show results for the SIM metrics and state of the art (top-1 model is bold)	91
5.15	Benchmark on saliency models. We show results for the AUC-Judd metrics and state of the art (top-1 model is bold)	92

1 Introduction

Since the 50s of the last century and until a very few years ago, the usual field of advanced Artificial Intelligence (AI) was mostly the research laboratory and science fiction. With the exception of few cases, almost all systems with human-like intelligence have appeared in futuristic films or works such as those of Isaac Asimov. However, this landscape has changed radically in recent years.

One of the building blocks of AI is machine learning. It is becoming increasingly common for us to ask machines to teach themselves. We cannot waste time pre-programming rules to deal with the infinite combinations of input data and situations that appear in the real world. Instead of doing that, we need machines to be capable of self-programming, in other words, we want machines that learn from their own experience. The discipline of Machine Learning addresses this challenge. Today machine learning is more than ever within the reach of any programmer. To experiment with these services we have platforms such as IBM Watson Developer Cloud, Amazon Machine Learning, Azure Machine Learning, TensorFlow or BigML.

Understanding the learning algorithms is easy if we look at how we learn ourselves as children. Reinforcement learning consists of a group of machine learning techniques that we often use in artificial systems. In these systems, as in children, behaviors that are rewarded tend to increase their probability of occurrence, while behaviors that are punished tend to disappear. These types of approaches are called supervised learning, as it requires human intervention to indicate what is right and what is wrong (that is, to provide reinforcement). In many other applications of cognitive computing, humans, apart from reinforcement, also provide part of the semantics necessary for algorithms to learn. That is, humans are the ones who

really know if a document is a complaint, an instance, a claim, a registration request, a change request, etc. Once the algorithms have a set of training data provided by humans, then they are able to generalize and begin to automatically classify documents without human intervention.

Currently, it is these training restrictions or limitations of algorithms that largely limit their power, since good training datasets (often manually labeled by humans) are required for algorithms to learn effectively. In the field of computer vision, for algorithms to learn to detect objects in images automatically, they have to be previously trained with a good set of labeled images, such as ImageNet. On the other hand, artificial vision is constantly evolving thanks to the optimization of algorithms based on Deep Learning. Deep Learning is a technology that allows a computer to learn like a human being, which makes it easier for artificial vision systems to use more robust, effective learning methods that are very similar to those of the human brain.

Object recognition is one of the main objectives of computer vision. Initially it was based on hand-crafted features, Lowe developed an image feature he called SIFT, that became the basis for features in many object recognition algorithms [114], but in recent years the most successful methods are based on convolutional neural networks (CNNs) [99]. Object recognition is a crucial functionality in many real-world applications, including robotics, automatic health care, autonomous driving, smart mobile applications, etc. It is well-known that humans apply a saliency mechanism to efficiently focus on the main information in images. Recent developments of neural networks have been exploited to estimate high-quality saliency images (i.e. DeepGazeII [97], SAM-ResNet [30], SALICON [72, 175] and SalGAN [137]), however, there has only been little research on how saliency can be incorporated efficiently in feed forward neural networks. Therefore, in this thesis, we explore how saliency can be used in CNN-based object recognition.

1.1 Challenges in Saliency for Object Recognition

For humans, object recognition is an almost instantaneous, precise and extremely adaptable process, since we keep learning during all our life. However, we are seldomly aware of the time and energy spent on creating those neurological structures that make such a complex process possible. It is well known how the human brain lowers this complexity by filtering out part of the information and only processing those things that capture our attention [5, 11, 16]. This, combined with our biological predisposition to respond to certain shapes or colors, allows us to recognize in a

1.1. Challenges in Saliency for Object Recognition

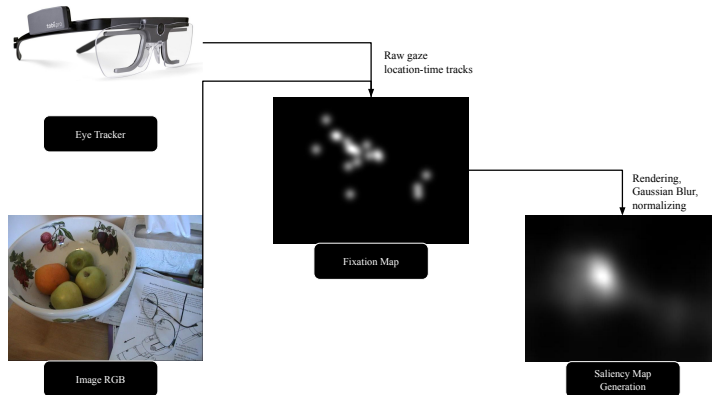


Figure 1.1: Example of saliency map based on eye fixations.

simple glance the most important sections from an image [76]. This is reflected by where we place our attention, and consequently on where we fix our eyes when an image is shown to us.

The most accurate way to record this behavior is to track eye movements while displaying multiple images [172]. Other methods require us to click with a mouse on the areas of interest [139, 171, 185]. Whatever the methodology, the results can be used to create an image of the same size as the original, where each pixel represents the probability that the subjects' eyes fixate on that pixel (see Figure 1.1). This way, the saliency map is created: a map that indicates the regions where humans are most likely to look, as this trait is crucial to improve our performance in image identification or tagging [11, 107, 205].

Given the huge effort to obtain saliency maps for large datasets, most of the research has been focused on how to obtain them automatically. In recent years, the most successful approaches have been based on convolutional neural networks (CNN) [29, 80, 137]. In this thesis, we are not focusing on the question how can we estimate saliency maps, but rather we are interested in the question if saliency maps can improve the object recognition task. In the following paragraphs we set out several research questions that we will address in this thesis.

One of the drawbacks of training deep CNN is that they require large amounts of data to train. For many applications, such as for example fine-grained object

recognition, this data might be hard to collect. As discussed above, saliency is used by the human visual system to focus on the most relevant visual information, and ignore superfluous background. Based on this observation, exploiting the use of saliency to prevent overfitting on small datasets is an interesting research directions. The saliency mechanism could help the network to focus on those parts of the image that are relevant and at the same time to ignore those parts of the images that are background, and therefore not expected to contribute much to the object recognition task. Based on this observation, we define our first research question to be:

- To what extent saliency maps can be exploited to improve the training of an object recognition model when only scarce training data is available ?

One drawback of networks, that use saliency maps to improve their object recognition performance, is that they require saliency maps as an input. This is typically done by training a separate saliency network. This saliency network requires ground truth data saliency maps to be trained. If it would be possible to train the saliency estimation branch within the object recognition network this would be highly beneficial. In this case, there is no need to compute the saliency map using a separate network. This would circumvent the need for the collection of large saliency datasets to train the network. Based on this reasoning, we explore ways to train saliency estimation directly within the object recognition network in an end-to-end way. Therefore, we define our second research question as:

- Can saliency be learned in an end-to-end sense within an object recognition pipeline based on a convolutional neural network ?

Finally, as a result of the research based on the two previous research questions, we propose an architecture that produces saliency maps that improve object recognition results. These saliency maps are a byproduct of the object recognition pipeline, and for their training no ground truth saliency maps are required. Traditional learned saliency estimation methods are based on ground truth saliency benchmark datasets. It would therefore be interesting to compare our saliency estimation approach, which does not require any saliency ground truth data, with methods that require saliency ground truth data on standard saliency estimation benchmark datasets. This leads to the third research question:

- What is the quality of saliency that is produced as a byproduct of object recognition ? How does is perform on benchmark saliency datasets ?

In the following section we set out our approach to address these questions, and the related research objectives and contributions.

1.2 Research Objectives

To address the research challenges identified in the previous section, we here define a set of research objectives.

The first research objective is defined as:

- **Demonstrate the importance of saliency maps for object recognition.** We are especially interested in object recognition based on convolutional neural network since these networks obtain the current state-of-the-art.

We first investigate to what extent the estimation of saliency can be exploited to improve the training of an object recognition model especially when scarce training data is available. To solve this problem, we design an image classification deep CNN that incorporates saliency information as input. This network processes the saliency map through a dedicated branch and uses the resulting characteristics to modulate the standard bottom-up visual characteristics of the original image input. We will refer to this technique as saliency-modulated image classification (SMIC). The main objective of the proposed method is to allow effective training of a detailed recognition model with limited training samples and to improve performance on the task, thus alleviating the need to annotate a large dataset. We evaluate our method on different datasets and in different settings, achieving considerable performance improvements when we take advantage of saliency data, especially when training data is scarce.

The second research objective of the thesis is given by:

- **Propose an architecture that does not require explicit saliency maps to improve image classification.** Instead these saliency maps should be learned implicitly, during the training of an end-to-end image classification task.

We start by addressing the main drawback of the above scenario, that is, SMIC requires an explicit saliency algorithm that must be trained on a saliency dataset. To solve this, we are going to implement a hallucination mechanism in order to eliminate the requirement to provide saliency images for training obtained using

one of the existing algorithms. In other words, we show that the explicit saliency branch that requires training using a saliency dataset can be replaced with a branch that is end-to-end trained for the image classification task (for which no saliency dataset is required). We replace the saliency image with the input RGB image. We then pre-train this network for the image classification task using a subset of the ImageNet validation dataset. During this process, the saliency branch will learn to identify which regions are the most discriminatory. In a second phase, we initialize the weights of the saliency branch with these previously trained weights. Then we train the system end-to-end on the fine-grained dataset using only the RGB images. The results show that the saliency branch significantly improves fine grain recognition, especially for domains with few training images.

The third research objective of this thesis is given by:

- **Demonstrate that it is possible to obtain accurate saliency maps without any ground truth saliency data.** Additionally, propose a fast and more realistic computation of the center-bias in an unsupervised manner. We show that the center-bias improves in most datasets where the center-bias is more present.

To do this, we evaluate the precision of the saliency maps that occur as a secondary effect of object recognition, based on the network which has been designed in this thesis. Furthermore, we also evaluated the use of supervised and unsupervised center (CB) bias in our setting. We show that CB improves in most of the datasets where CB is more present. We perform these experiments on several standard benchmark datasets for saliency estimation.

1.3 Contributions and Outline

The main contributions of this dissertation are:

- **We demonstrate that the use of saliency improves the classification task based on forward convolutional neural networks (Chapter 3).** The gain due to saliency is especially notable for domains with scarce data.
- **We address the major drawback of saliency-modulated image classification (SMIC) (Chapter 4).** We implement a hallucination mechanism in order to remove the requirement for providing saliency images for training obtained using an existing saliency algorithms.

- **We demonstrate with several experiments that our approach is able to generate accurate saliency maps (Chapter 5).** We achieve competitive results when compared with supervised methods. Our saliency maps are a side-effect in an object classification tasks. We also investigate the use of the center-bias within this framework.

The aforementioned contributions have been presented at conferences and published in a scientific Journal. More details of such publications are included in the Chapter 6.

The thesis is organized as follows. In chapter 2 relevant works related to the proposed approaches are summarized into three sub-fields: *i*) representation learning; *ii*) visual saliency; and *iii*) fine-grained object recognition. Chapter 3 describes our approach to use saliency in order to improve classification accuracy for fine-grained object recognition in domains with scarce training data. In Chapter 4 we improve this approach, and describe a method that can hallucinate saliency maps for fine-grained image classification. Then, in Chapter 5 we evaluate the quality of saliency maps that are computed as a side-effect of our object recognition architecture. We show that these maps can obtain excellent results on saliency benchmark datasets. Finally, in Chapter 6 the thesis is concluded and we present future work.

2 Related Work

In this chapter we discuss the related work. We start with a brief overview of deep neural networks. We then discuss the literature on visual saliency, and finally discuss the literature on fine-grained object recognition.

2.1 Learned Representations

The complexity of the information processing task depends on how the information is represented. To this end, many approaches to representation learning, whether linear or non-linear, supervised or unsupervised, "shallow" or "deep", have been developed to understand the intrinsic structure of data. In this context, feature learning or representation learning [7], is a set of techniques that allows a system to automatically discover the representations required for the detection or classification of features from raw data. In particular, deep architectures have provided the best results in many tasks such as image classification, object detection, and speech recognition [228].

Many data representation learning methods have been proposed in the last hundred years in order to learn low dimensional representations of data, for example, K. Pearson in 1901 [48] proposed a linear projection, principal component analysis (PCA), while linear discriminant analysis (LDA) was proposed by R. Fisher in 1936 [45]. PCA and LDA are both the earliest data representation learning algorithms (unsupervised and supervised methods respectively). Since 2000, the machine learning community launched the research on manifold learning, which

is to discover the intrinsic structure of high dimensional data [6, 155].

In 2006, Hinton and his co-authors successfully applied deep neural networks to dimensionality reduction, and proposed the concept of **Deep Learning** [66, 67]. Deep Learning (DL) is an area of machine learning that is based on the usage of hierarchical structures and algorithms inspired by the human brain which provides a multidimensional space for learning multiple levels of representations in order to model complex relationships among data [33, 35, 57]. The key aspect of deep learning is that the feature representation in the network's layers is not manually designed, but they are automatically learned from data during the training process. For example in the Figure 2.1, it is shown how a deep learning network can represent the concept *person* by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges.

The most simple example of a deep learning model is the deep feedforward network represented as a multilayer perceptron (MLP). An MLP implements a function mapping a set of input values to output values [57]. This function is formed by many simpler functions. We can think of each layer as a different mathematical function which provides a new representation of the input. Another perspective on deep learning is that depth enables the computer to learn a multi-step computer program, where each layer can be thought of as a state of the computer's memory executing a set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. These sequential instructions offer a great advantage, because subsequent instructions can rely on the results of earlier instructions. According to this view of deep learning, not all the information in the activations of a layer necessarily encodes factors of variation that explain the input. The representation also stores state information that helps to run a program that can make sense of the input. This has nothing to do with the content of the input specifically, but it helps the model organize its processing.

Specifically in computer vision, which is a field that focuses on the understanding of information presented in the form of image or video data, deep neural networks have shown impressive results. However, the limitation of MLPs in this case was obvious: due to their architecture, they were flattening the pixels and thus discarding the image structure. Therefore, in order to take into account the spatial relationship between pixels, a different strategy was required and, as a result, convolutional neural networks (CNN) have been introduced [99] (more details in section 2.1.2). To have a better picture of the recent advances with CNN, it is enough to mention the evolution of some specific tasks, ranging from 'simple' ones such as image classification/object to more complex applications for astronomy problems

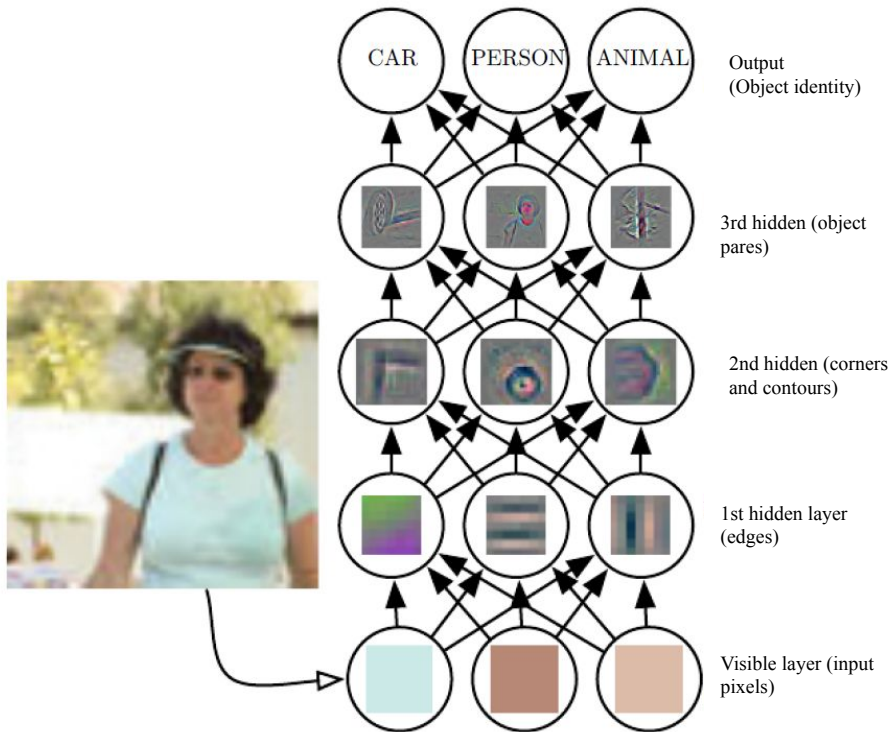


Figure 2.1: Illustration of a deep learning model from [57].

or self-driving cars [99].

One of the most common tasks in computer vision is image classification. It was the ImageNet competition [158] that was responsible for the emergence of more precise models and a better understanding of the convolutional networks themselves. A fortuitous event demonstrated that those models trained for this competition could be reused in other scenarios and worked quite well, sometimes even surpassing the models trained specifically for the tasks [224]. This is due to the large variety of images (over a thousand categories), which are displayed in the dataset provided for the ImageNet competition. As a result, ImageNet-trained models see the world in much the same way as the human visual system. It is not so surprising then, that after a few decades of trying to develop better networks (to improve performance in visual tasks) they converged to architectures that potentially

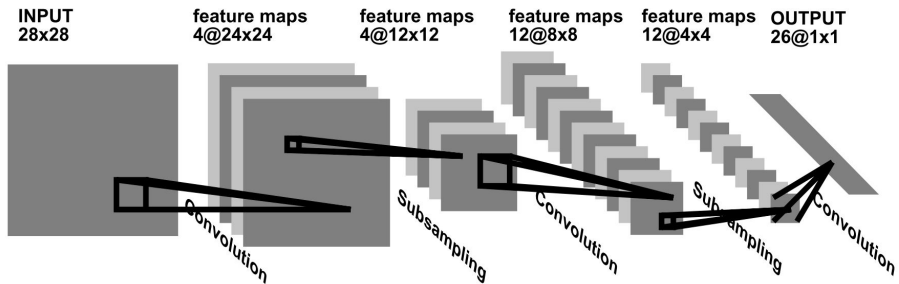


Figure 2.2: Convolutional Neural Network for image processing, e.g., handwriting recognition [99].

function in a similar way to the visual cortex of some primates as found in [23, 124].

2.1.1 Convolutional Neural Networks

The first CNN based architecture was proposed in 1995 by LeCun *et al.*[99]. Figure 2.2 shows the convolution neural network architecture from [99], where the input of the architecture is a 28×28 gray-scale image, then, such data is forward propagated through the five convolutional neural layers till it gets the size of $26 \times 1 \times 1$. The data in each CNN layer is termed a **feature map** and its size, for example, in the first CNN hidden layer, "4x12x12" is interpreted such as 4 filters with the feature map size of 12×12 . Even though, the CNN architectures were proposed in the last century, the success of this approach was noticed, principally, with the AlexNet [94] proposal. This CNN architecture is composed by five convolutional and three fully-connected layers, as observed in Figure 2.3, such a design was sufficient to win the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012) competition achieving the test error rate of 15.3%, the second best competitor scored 26.2%.

Since **AlexNet** was introduced in 2012, many other CNN-based architectures have been proposed for different computer vision applications, e.g. edge detection [204], segmentation [220], image recognition [162], to name only some of them. However, just a few of these architectures have been efficiently designed to be replicated in other computer vision tasks. Among the most popular networks are: VGG [162], U-Net [154], ResNet [62], Inception v3 [170], Xception [28].

The Visual Geometry Group (**VGG**) at the University of Oxford. Among them,

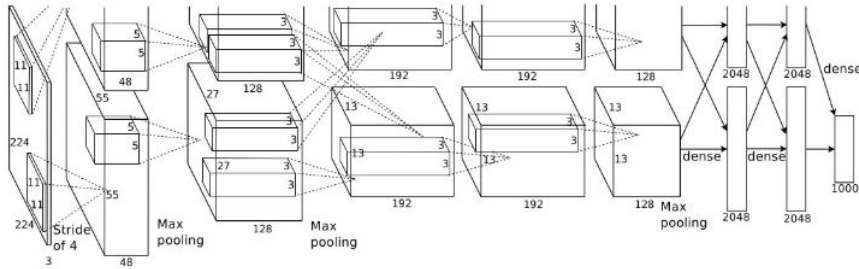


Figure 2.3: Illustration of the architecture of the AlexNet CNN introduced in [1].

VGG16 and VGG19 are the most used in different computer vision and image processing tasks [162]. For instance, the VGG16 version can be appreciated in Figure 2.4. This architecture is designed by blocks of convolutional and fully-connected layers, specifically, composed by 13 convolutional and 3 fully-connected (MLP) layers. VGG architecture scored the second best mark in the classification task of ILSVRC-2014 competition.

One year later, the Deep Residual Networks (**ResNet**) have been proposed in [62]. Like VGG, the ResNet family has also different versions according to the depth of its layers (ResNet50, ResNet100, ResNet152, etc.), with ResNet50 being one of the most used in the literature. Although the deeper neural network is, the more difficult its training is (due to vanishing gradient problem), the solution to overcome this limitations resides in the use of skip connections (termed in the paper as "shortcut connections"). Skip connections help retaining the correlation structure across gradients [4] (thus alleviating the aforementioned problem) and thus determining an improved training and an improved performance as a result. Skip connections are additional connections between nodes in different layers that can skip several layers which form a residual block. In this configuration, the output of the current residual block is summed with the identity mapping from the previous residual block, this process being repeated several times along the network's depth (see Figure 2.5).

2.1.2 Neural Networks for Scarce Data Domains

One of the themes of this thesis is the training of networks for scarce data domains. Few-shot learning aims to create models for which very few labeled samples are available.

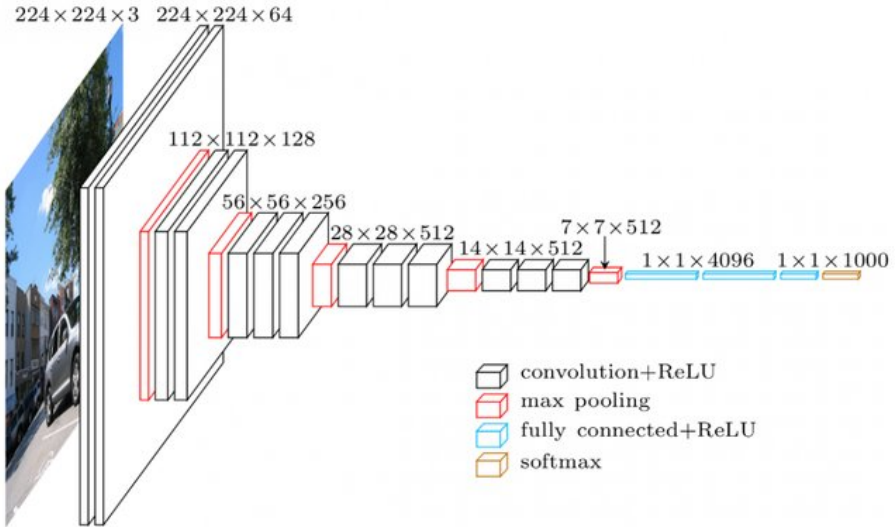


Figure 2.4: The VGG16 architecture [113].

Early work on this topic is attributed to Fei-Fei et al. [42] who showed that, taking advantage of previously learned categories, it is possible to learn new categories using one or very few samples per class. More recently, [92] proposed a conditional distance measure that takes into account how a particular appearance model varies with respect to every other model in a model database. The approach has been applied to one-shot gesture recognition. Nowadays, several deep learning-based approaches have emerged to address the problem of few-shot learning. We can identify three main strategies.

One family is based on metric learning. In [186], the authors proposed a framework that trains a network to map a small labeled support set and an unlabeled example to its label. An extension of this idea is presented in Prototypical networks [164], but in this case each class in the support set has been substituted by a 'prototype' (computed as the mean of the samples in the corresponding class), to which each sample is compared.

A second family of approaches is based on meta-learning, i.e. learning a model that given a few training examples of a new task tries to quickly learn a learner model that solves this new task [128]. In [147], the authors propose an LSTM-based meta-learner that is trained to optimize a neural network classifier. The meta-

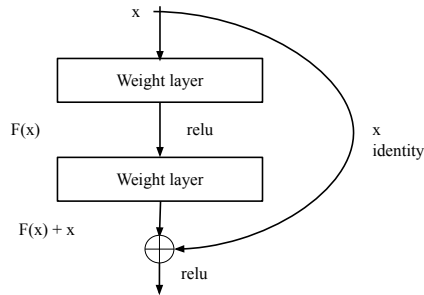


Figure 2.5: Residual block with skip connection [62]

learner captures both short-term knowledge within a task and long-term knowledge common among all the tasks.

Finally, the third family of approaches is based on data augmentation for data-starved classes. In [61], the authors propose a way to increase ("hallucinate") the number of samples for the classes with limited data. Their method is based on the intuition that certain aspects of intra-class variation generalize across categories, like for instance pose transformations. In practice, for data-rich classes, they use a neural network to learn transformations between pairs of samples and this transformation is later on applied on the real samples from data-starved classes to generate synthetic ones, thus increasing the population of these classes. For the same purpose (i.e. data augmentation for data-starved classes), in [37] the authors propose an attributed-guided augmentation approach which learns a mapping that allows the creation of synthetic data by manipulating certain attributes of real data. Thus, the newly created data presents attributes based on user-defined criteria (values). Instead of performing the data augmentation in image space, they perform it in feature space. This idea is further extended in [110], where the authors use a deep encoder-decoder architecture to generate feature trajectories by exploiting the pose manifold in terms of pose and appearance.

2.2 Visual Saliency

Visual saliency has long been one of the most studied problems in neuroscience, psychology, and computer vision and can be defined as "the ability that makes

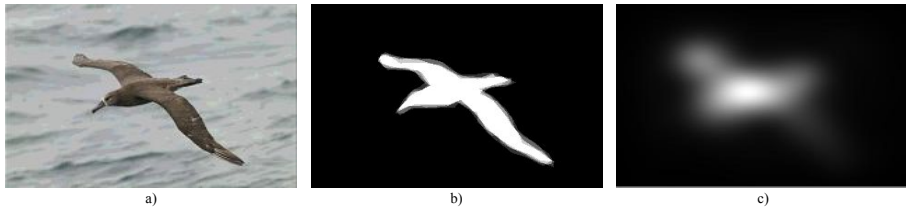


Figure 2.6: Example of salient object in natural image. a) original image; b) ground truth ; c) example saliency detection results.

some elements of the world stand out from its neighbors and grab our attention immediately." [73]. In other words, our attention is selective and our brain highlights objects that contrast with other elements [181]. However, simultaneously identifying each and every interesting target in the visual field is computationally complex, making it a daunting task for even the most sophisticated biological brains, let alone any existing computer [178].

Saliency is generally known as local contrast [75], which generally originates from contrasts between objects and their surroundings, such as differences in color, texture, shape, etc. This mechanism measures intrinsically outgoing stimuli to the vision system that primarily attract human attention in the initial stage of visual exposure to an input image [60].

To quickly extract the most relevant information from a scene, the human visual system pays more attention to the highlighted regions, as seen in Figure 2.6. Research on computational saliency focuses on the design of algorithms that, like human vision, predict which regions of a scene stand out. As a definition, visual saliency is the perceptual quality that makes an object, person, or region of pixels stand out in relation to its neighbors to attract our attention [76].

The intermediate and upper visual processes can automatically judge the importance of different regions of the image and carry out detailed processes only on the "salient object" that mainly relates to the current task, while neglecting the remaining regions of "background" [15]. Figure 2.6 shows some examples of natural images. As seen in Figure 2.6 (c), the bird is the one that attracts the most visual attention and, therefore, they are considered as salient objects. On the other hand, Figure 2.6 (b) shows an example of "saliency map" detection.

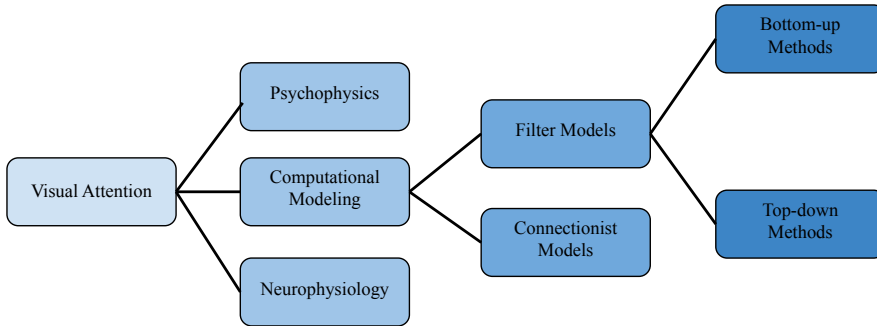


Figure 2.7: Saliency detection methods ontology [15]

2.2.1 Saliency Methods

Saliency detection methods can be grouped according to the model inspiration source. For instance, Itti’s approach is referred as a biological inspired method. Such methods explore peculiarities of human vision and attention operation and try to mimic the processes taking place while a human observes a scene.

From the perspective of computer vision, the methods of saliency detection are broadly categorized into two major groups, namely the bottom-up methods and the top-down methods (see Figure 2.7. Besides that, more methods using unconventional models and features have also been proposed in recent years.

Bottom-Up Methods

The bottom-up saliency methods describe the attention distribution of a visual stimulus (image) [26, 167, 206, 229], in form of attention map. In this case, the attention is driven by low-level features such as color, contrast, and therefore they are task-independent

Initial work on computing saliency was due to Itti, Koch and Niebur [76], proposed one of the first biologically motivated computational models for saliency estimation. Their saliency map was inferred from multi-scale representations of color, orientation and intensity contrast. Saliency research was propelled further by the availability of large datasets which allowed for direct comparison of methods and enabled the use of data-driven methods based on machine learning algorithms.

Rahtu *et al.* [145] proposed a model using a contrast of a sliding window over the input image. The resulting saliency map was then used in a Conditional Random Field (CRF) model to define a segmentation approach based on energy minimization, which aims to recover well-defined prominent objects. All these maps were fused in a unique saliency map using winner-take-all mechanisms [145]. This framework had inspired many models such as [12, 153, 213], mainly varying on the feature extraction part (either handcrafted or trained). For instance, some unsupervised models such as the work proposed in [18] used a dictionary of images in order to train sparse priors which were learned with a feature extractor filters. Later on, in [17], the authors presented a combination between feature extraction from Itti-Koch-Niebur model (IKN) and Attention based on Information Maximization model (AIM). The resulting model was an architecture with cells and connectivity reminiscent of that appearing in the visual cortex. Similarly, [177] proposed a contextually-modulated saliency model, which was based on task priors when observing real scenes and predicts the image regions likely to be fixated by human observers performing natural. Finally, Murray *et al.* [130] show how a model of color appearance in human vision can be generalized to obtain a saliency model.

Since the 2010s, more advanced models, and especially the graph based models, have been introduced to saliency detection, which have greatly improved the overall detection accuracy [95]. It is also worth noting that the majority of conventional low-level feature based saliency detection methods were proposed during this period. For example, Jiang *et al.* [78] formulated saliency detection via absorbing Markov chain on an image graph model and the absorbed time from each transient node to boundary absorbing nodes is computed. Thus, salient objects can be consistently separated from the background when the absorbed time is used as a metric. On the hand, Li *et al.* [102] proposed a novel approach that takes advantage of both region-based features and image details. The first step consists of optimizing the image boundary selection by the proposed erroneous boundary removal, which is followed by a second step consisting of the foreground saliency estimation. In [141], Perazzi *et al.* defined a conceptually algorithm for contrast-based saliency estimation. Their algorithm consists of several steps. First, they decompose a given image into compact, perceptually homogeneous elements that abstract unnecessary detail. Then, they compute two measures of contrast that rate the uniqueness and the spatial distribution of these elements. Finally, from these elements they derive a saliency measure that produces a pixel-accurate saliency map which uniformly covers the objects of interest and consistently separates foreground and background. Wei *et al.* [198] proposed a novel saliency measure called geodesic saliency, by exploiting two common background priors: boundary and connectivity. In the work of Yang *et al.* [208], the authors described an approach by exploiting contrast,

center and smoothness priors. First, they computed an initial saliency map using contrast and center priors, applying the convex hull of interest points to estimate the center of the salient object rather than directly use the image center. Second, they exploited the graph-based manifold ranking to extract foreground queries for the final saliency map, in which the four image boundaries are used as background prior knowledge.

Recent models (e.g. ML-Net [29], such as SAM [30], DeepGazeII [97], SalGAN [137]) use fixation data from image saliency datasets (i.e. that provide eye tracking data) as ground truth for learning the saliency map with CNN architectures. These models usually train a neural network that focuses on the most salient regions of the input image to iteratively refine the predicted saliency map. ML-Net learned a prior map based on the common ground truth saliency maps, acting as a mask. This is multiplied by the output map of the network. For the case of DeepGazeII they sum a probability distribution (baseline of fixations) over the image. Instead, SAM utilizes an LSTM and trains a set of gaussian parameters acting as an attentive mechanism to the final map, which finetuned with human fixation density maps. Finally, SalGAN uses an autoencoder architecture, which is trained with prediction in combination with an adversarial loss.

Top-Down Methods

The top-down saliency detection methods emerges from those regions that consciously attract users' attention according to a specific visual task [129], i.e. they are task dependant. Thus, top-down attention is slow and deliberative with variable selection criteria depending on the task.

In the early 2000s, Itti and Koch [75] proposed the idea of top-down influence to better estimate the saliency in specific tasks. They considered that there was a link between visual attention and eye movement. Thus, it was necessary to combine eye movement with a computational model to study the human visual system.

Supervised learning approaches are commonly used in detecting image saliency. In their work, Yang *et al* [209], proposed a novel model that jointly learned a Conditional Random Field (CRF) and a visual dictionary. Their proposal thus produced clear saliency maps by incorporating local context information. Improved results were obtained by updating the dictionary under the CRF supervision. In Lu *et al* [115], saliency of salient seed locations was propagated through the graph via a diffusion process. Unlike previous heuristic approaches to seed selection, an optimal set of salient seeds is learned using a large margin formulation of the discriminant

saliency principle. Mai *et al* [120] presented an approach based on Conditional Random Field (CRF) framework for saliency aggregation that not only models the contribution from individual saliency map, but also takes into account the spatial relation among pixels. Moreover, Xu *et al.*[207] proposed a deep unified CRF saliency model that formulates messages passing with CRFs for joint feature and prediction refinement. Tong *et al* [176] proposed a bootstrap learning model for salient object detection. The strong saliency model is constructed based on the Multiple Kernel Boosting (MKB) algorithm, which combines all the weak classifiers into a strong one using the Adaboost algorithm.

Since 2013, thanks to the tremendous success of deep learning and other high-level feature extraction techniques, more learning-based methods emerged with significantly improved performance. In [79], the authors formulated saliency map computation as a regression problem. Their method, based on multi-level image segmentation, used the supervised learning approach to map the regional feature vector to a saliency score. Saliency scores across multiple layers were finally fused to produce the saliency map. Kim *et al* [86] introduced a novel technique to automatically detect salient regions of an image via high-dimensional color transforms. Their main idea was to represent a saliency map of an image as a linear combination of high-dimensional color spaces where salient regions and backgrounds can be distinctively separated. This is based on an observation that salient regions often have distinctive colors compared to the background in human perception, but human perception is often complicated and highly nonlinear. Wang *et al* [189] presented a Deep Neuronal Network (DNN) for saliency detection from both local and global perspectives. In the local estimation stage (DNN-L), they estimated local saliency by learning image feature from local contrast, texture and shape information. In the global search stage (DGG-G), they exploited the complex relationships among global saliency cues and predicts the saliency value for object region. Zhao *et al* [225] proposed a multi-context DNN for saliency detection. The global context was utilized to model saliency in full image, while the local context was used for saliency prediction in meticulous areas.

Li *et al.* [104] proposed a novel method for saliency detection using a CNN which has fully connected layers on its top, responsible for feature extraction at three different scales. Fusing the saliency maps corresponding to these scales into a single one, outperforms other methods which gearates saliency maps from a single segmentation. Chen *et al.* [26], presented a novel approach called Deep Image Saliency Computing (DISC) using both the coarse- and fine-level observations in order to learn the saliency representation in a progressive manner. They used a two-stacked CNNs, one for generating a coarse-level saliency map (image level) and

the other one for generating a more accurate saliency map (by focusing on the local context).

On the other hand, Murabito *et al.* [129] presented an approach that differs from others since the only top-down signal introduced in their training is a class-agnostic classification loss, i.e. their maps were able to highlight those areas which are relevant for classifying generic images. A similar work is the one of Almahairi *et al.* [2], where they introduced a Dynamic Capacity Network (DCN). The authors combined two types of sub-networks: the first is a low-capacity sub-network and the second is a high-capacity sub-network. The low-capacity sub-network is applied across most of the input, but also provide a guide to select a few regions of the input on which to apply the high-capacity sub-network. The selection is made using a novel gradient-based attention mechanism, that efficiently identifies input regions for which the DCN's output is most sensitive.

Other Methods and Current Trends

Besides the two main categories previously analyzed, in this section we review some other approaches based on hybrid methods, that take advantage of both the high detection accuracy of top-down features extraction and the high detection efficiency of bottom-up methods. Additionally, we will review some recent trends represented by salient object detection. Finally, we will mention some applications of saliency detection.

Hybrid methods are biologically motivated and the research of Melloni and *et al* [125], who based their study on a functional magnetic resonance imaging analysis, found evidence of a hierarchy of saliency maps in human early visual cortex (V1 to V4) and identified the region where bottom-up saliency interacts with top-down control. Following this findings, Borji and *et al* [10] proposed a method where they combined low-level features such as orientation, color, intensity, saliency maps of previous best bottom-up models with top-down cognitive visual features (e.g., faces, humans, cars, etc.) and learned a direct mapping from those features to eye fixations using regression, SVM, and AdaBoost classifiers. Shariatmadar *et al* [161] proposed a hybrid method for extracting relevant regions of man-made objects. Top-down path is implemented by extracting features characteristic to edges and corners; bottom-up path is implemented by the response of Gabor filters of different orientations. Finally, these maps are linearly combined in order to generate the saliency maps. In [118], Mahdi and *et al* presented an exhaustive study of a computational saliency model based on pre-trained deep features to predict human fixations. The bottom-

up path was modeled by the deep features extracted from the convolutional layers of a two-scale CNN, while the top-down path was represented by the deep features extracted from the whole network. The combination of the two paths is weighted using a center bias map. Different from this late-fusion approach, in [191] Wang *et al* integrated both top-down and bottom-up inference paths in an iterative and collaborative manner.

In the last few years, the research on saliency prediction has shifted its focus from attention maps to object detection. Different from the study of attention maps which has its roots in neuroscience and cognitive psychology, salient object detection (SOD) is motivated by computer vision applications targeting object-level processing. Although the basis for salient object detection could be traced back to the work of Liu *et al* [111], the recent advances in deep learning have revitalized this research direction, where most approaches are built upon fully convolutional neural networks (FCN), U-NET or feature pyramid networks (FPN) as their basic structures. Salient object detection first locates and identifies the object/region in the image, and then segments it from its background. For this purpose, a lot of models have been proposed, which have achieved a good performance in simple images containing either a single object [105, 106, 132] or multiple objects [40, 218], which have been also extended to salient-instance segmentation [41, 201]. Some interesting applications of SOD are salient object subitizing, i.e. instant judgement of the number of objects in the image [63, 216] and saliency rank of objects defined by the order in which a person attends the objects presented in an image [163]. As depth cameras (such as Kinect or RealSense) become popular in the last few years, RGB-D object detection attracted more and more research interest. While early methods were based on a simple deep fusion scheme [144], most recent approaches are built upon a more sophisticated distillation mechanism [116, 142], in order to transfer the knowledge between the RGB and Depth modalities, or gating mechanism [226].

All the saliency methods reviewed so far are supervised, i.e. they require a saliency map ground truth in order to train the deep neural network. This represents a significant limitation nowadays since there are applications which generate a huge amount of data (e.g. self-driving cars, multimedia). Computing the saliency maps for all these images is an intractable problem and thus could limit the generalization capabilities of the method. Therefore, another recent effort in saliency research is towards unsupervised saliency detection. For instance, Zhang *et al* [219], formulate the problem of unsupervised saliency detection as a learning process from multiple noisy maps generated by various conventional methods. They propose an end-to-end deep learning framework which consists of a saliency prediction module and an

explicit noise modeling model, which work collaboratively and are jointly optimized. On the other hand, Sun *et al* [169] generate saliency maps in an unsupervised manner by exploiting the expectancy-mismatch hypothesis: using a pre-trained network, they provide a 'reference pattern' which is in conflict with the current output. Backpropagating this error to a semantically meaningful convolutional layer, they obtain, at the end of the training process, the saliency map. Despite being a simple approach, they obtain competitive results, when compared with supervised methods. Finally, another approach is presented in Palazzo *et al* [136] where they consider saliency maps generation in a multi-modal context, by modulating the deep visual representation of an image with the neural activity captured by an EEG device while the subjects look at images.

2.2.2 Saliency for Image Classification

The vast majority of saliency methods previously reviewed are evaluated on the task of how accurate their generated saliency maps are.

Therefore, it was raised the question of whether saliency is also important for other related tasks such as object recognition and object tracking [59]. This is also the purpose of [46], where the authors investigate to what extent saliency information can be exploited to improve object recognition when the available training data is scarce. The authors designed a two-branch image classification deep network, where one of the branches takes saliency information as input. The network processes the saliency through the dedicated branch and uses the resulting saliency features to modulate the visual features from the standard RGB branch, thus forcing the upper layers to focus on the relevant parts only. In the same line, [129] learned to generate saliency maps from RGB images, but in this case their method is supervised.

2.2.3 Link between attention and saliency

Attention is widely known to be fundamental to perception, the term being often used to mean very different things [58, 180]. The most important theories of attention relate it to planned or executed eye movements [127]. This contrasts with the notion of attention as a gain control process that weights the information carried by different sensory channels. Also, attention influences the processing of visual information even in the earliest areas of primate visual cortex [84, 126, 135].

There is converging evidence that the interaction of bottom-up sensory informa-

tion and top-down attentional influences creates an integrated saliency map, that is, a topographic representation of relative stimulus strength and behavioral relevance across visual space [74, 125, 180]. This influence seems to shape an integrated saliency map, that is, a representation of the environment that weighs every input by its local feature contrast and its current behavioral relevance [139, 143, 152]. This generated saliency map provides a coding scheme to process the most relevant information in the sensory input to the visual system and thus integrate large volumes of information. However, by completely integrating bottom-up sensory information and top-down attentional influences it equates the absence of attention with low stimulus power [180]. This could explain why highly prominent stimuli will be processed even in the absence of attention. On the other hand, the inherent low prominence will often prevent perceptual representations for some parts of complex natural scenes [65, 134, 180]. Although, exceptions seem to exist for basic categorizations, that is, a recovery of the 'gist' of natural scenes, as we can observe in Figure 2.8.

It is known that the brain areas that are responsible for providing the correct orientation for the top-down process are strongly related to those areas responsible for the execution of eye movements [123, 160, 180]. Bringing together bottom-up stimulus aspects (that are often responsible for automatic attentional allocation) and top-down influences (that reflect voluntary attention), a global map representing stimulus saliency (that is modulated by the current behavioral state of the organism) can be computed. This can provide a unified framework for interpreting future findings on attentional effects and their close integration with sensory information processing [180].

2.2.4 Center bias

Observers, when looking at the computer screen, have been found to have a marked tendency to look at the center of the screen [34, 148, 171, 185]. Usually this happens, because the characteristics of the images / scenes tend to be skewed towards the center of the natural images and the fixations are correlated with image features [122, 139, 140, 156, 157, 172].

Although this tendency to look towards the center is well documented, the reasons for this bias are not yet clear [171, 172]. Center bias offers an interesting opportunity to explore not only the underlying mechanisms responsible for this trend, but also the degree to which fixations are determined by the image characteristics present in the scenes [31, 39, 183]. In the state of the art, two possible

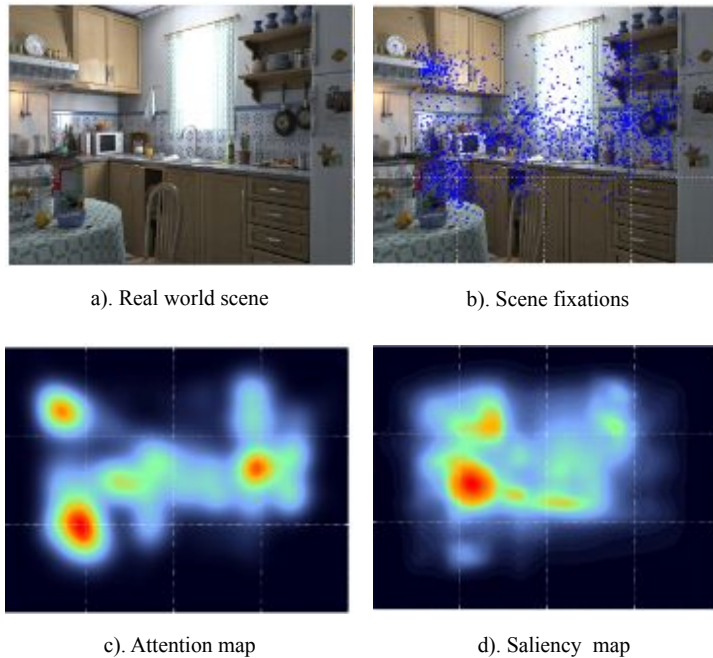


Figure 2.8: An example of Attention map and Saliency map from a real world scene. [65].

explanations have been found for the tendency of human observers to look at the center of scenes rather than at the periphery. These will now be discussed in the next paragraphs [171, 212].

First, center bias may be the result of certain biases in small-amplitude eye movements over large-amplitude (saccadic) eye movements as explained in [44, 98, 151, 173]. Given this tendency to perform small saccades, the fact that scene visualization experiments usually use a centrally located pre-trial fixation marker results in the central bias observed in fixation distributions [43, 100, 140, 212].

And, second, it was assumed that the bias arises from selecting image features for fixation, which are often centrally biased in the scenes [139, 149, 172]. Several recent studies have shown that the locations selected for fixation by human observers tend to correlate with low-level image features in the scene [74, 139, 140, 150, 171,

172]. In particular, fixated locations tend to have higher-than-average contrast and edge information [3, 140, 149]. For instance, most real images frame the scene, meaning that the relevant or salient part is in the center of view in photographs. Non-salient/non-popout stimuli [9, 171, 185] has been shown to promote center biases, as participants do not have any region to attend to, especially if the task sometimes involves centering the gaze on the image.

These center biases have an influence on how to evaluate saliency models upon predicting fixations [14, 21], as these fixations are accounted while are not specific to image saliency. This bias is in part because photographers tend to place objects of interest at the center of the viewfinder. Thus, if fixations and features correlate, a centrally biased distribution of features in scenes would result in the observed central biases in human fixation distributions.

2.2.5 Datasets for Saliency Estimation

Since the available eye movement datasets have different statistics, types of stimuli and numbers of subjects, here we exploit three categories of reference datasets for a fair comparison of models, which are divided into i) real images scenes, ii) natural scenes and iii) synthetic images [9, 10, 20, 22].

i. Real images scenes:

The first dataset, **MIT1003** created by Judd and *et al* in [81], contains 1003 images from Flickr and LabelMe datasets and eye-tracking data recorded by fifteen users who viewed these free images. These users were men and women between 18 and 35 years old. Of these users, two were project researchers and the others were only viewers [54, 159, 184]. An eye tracker recorded the trajectory of users' gazes on a separate computer as they viewed each image at full resolution for 3 seconds separated by 1 second of viewing a gray screen. To ensure high-quality tracking results, they checked the camera calibration every 50 images. To obtain a continuous prominence map of an image from a user's eye-tracking data, the authors construct a Gaussian filter through the user's fixation locations. Furthermore, the authors generated a saliency map of the average locations set by all viewers [54, 55, 184]. On the other hand, the longest dimension of each image was 1024 pixels and the other dimension ranged from 405 to 1024, with the majority being 768 pixels [54, 159, 184]. There were 779 landscape images and 228 portrait images [96, 97].

The second dataset, **Toronto** defined by Bruce *et al*. [18], is the most widely-used dataset for saliency model evaluation, according to [10, 15, 25, 190]. This

dataset contains a variety of 120 images of interior and exterior scenes, some with highlights and others without anything of interest. The eye tracking device consisted of a standard non head-mounted device. The parameters of the setup are intended to quantify saliency in a general sense based on stimuli that one might expect to encounter in a typical urban environment. Data was collected from 20 different subjects for the full set of 120 images [184, 213].

ii. Natural scenes:

The most used dataset for natural scenes is KTH, which was proposed by Kootstra *et al.* [89]. This dataset contains 99 photographs from 5 categories, i.e. animals, human-actions, buildings, flowers and nature. Each photograph was observed by 31 subjects [103]. Here, human fixation data was recorded during an eye tracking experiment using the EYELINK head-mounted eye tracking system (SR research). Then, the images were displayed full-screen with a resolution of 1024 by 768 pixels on an 18 inch CRT monitor of 36 by 27 cm at a distance of 70 cm from the participants. The eye tracker was calibrated using the EYELINK software. The calibration was verified prior to each session, and recalibrated if needed. The participants were asked to free view the images [89, 103].

The observers were not given a specific task, since the interest was to observe the bottom-up components of visual attention. The experiment was carried out by 31 students from the University of Groningen (The Netherlands) [119].

iii. Synthetic images:

CAT2000p is a training subset of "Pattern" images belonging to the larger dataset CAT2000 [13], containing 200 psychological patterns which have often been used for evaluation of bottom-up saliency models, mainly in behavioral studies including pop-out, conjunction, search asymmetry, etc. This subset provides eye movement data with psychophysical/synthetic image patterns during 5 sec of free-viewing.

Another synthetic dataset is **SID4VAM** proposed by Berga *et al.* [9], which contains fixations collected from 34 participants grouped in a collection of 230 images. The images were displayed in a resolution of 1280×1024 px and fixations were captured at about 40 pixels per degree of visual angle using SMI RED binocular eye tracker. The dataset has been split in two tasks: Free-Viewing (FV) and Visual Search (VS). For the FV task, participants had to freely look at the image during 5 second. Instead, for the VS task, participants had to visually locate the area of interest.

2.3 Fine-grained Object Recognition

Fine-grained object recognition aims to classify subclasses belonging to the same category [46, 52, 82]. Examples of fine-grained datasets include natural categories such as flowers [133], birds [199], dogs [85] and man-made categories such as cars [90], among others. The problem of fine-grained object classification is difficult because the differences between subclasses are often subtle and expert labelers, with knowledge of the discriminating attributes, are needed for the collection of datasets [46, 179].

Most of the state of the art general object classification approaches [93, 188] have difficulties in the fine-grained recognition task, which is more challenging due to the fact that basic-level categories (e.g. different bird species or flowers) share similar shape and visual appearance. One reason for this could be attributed to the popular codebook-based image representation, often resulting in the loss of subtle image information that is critical for the fine-grained task [64].

A first group of approaches on fine-grained recognition operate on a two-stage pipeline: first detecting some object parts and then categorizing the objects using this information. The work of [71] first localizes a set of part keypoints, and then simultaneously processes part and object information to obtain highly descriptive representations. Mask-CNN [197] also aggregates descriptors for parts and objects simultaneously, but using pixel-level masks instead of keypoints. The main drawback of these models is the need of human annotation for the semantic parts in terms of keypoints or bounding boxes. To partially alleviate this tedious task of annotation, [202] proposes a weakly-supervised approach based on the combination of three types of attention in order to guide the search for object parts in terms of 'what' and 'where'. A further improvement has been reported in [223], where the authors propose an approach free of any object / part annotation. Their method explores a unified framework based on two steps of deep filter response picking. On the other hand, [194] proposes an end-to-end discriminative feature-oriented Gaussian Mixture Model (DF-GMM) to learn low-rank feature maps which alleviate the discriminative region diffusion problem in high-level feature maps and thus find better fine-grained details.

A second group of approaches merges these two stages into an end-to-end learning framework which optimizes simultaneously both part localization and fine-grained classification. This is achieved by first finding the corresponding parts and then comparing their appearance [192]. In [203], their framework first performs unsupervised candidate-part discovery and global object discovery which

2.3. Fine-grained Object Recognition

Dataset	Year	Meta-class	# images	# Categories
Oxford Flowers [133]	2008	Flowers	8.189	102
CUB200 [187]	2011	Birds	11.788	200
Stanford Dog [85]	2011	Dogs	20.580	120
Stanford Car [90]	2013	Cars	16.185	196
FGVC Aircraft [121]	2013	Aircrafts	10.000	100
Birdsnap [8]	2014	Birds	49.829	500
NABirds [182]	2015	Birds	48.562	555
DeepFashion [112]	2016	Clothes	800.000	1.050
Fru92 [70]	2017	Fruits	69.614	92
Veg200 [70]	2017	Vegetable	91.117	200
iNat2017 [69]	2017	Plants/Animals	859.000	5.089
RPC [195]	2019	Retail products	83.739	200

Table 2.1: Benchmark datasets for Fine-grained classification from [196]

are subsequently fed into a two-stream CNN in order to model jointly both the local and global features. In [27], they propose an approach based on 'Destruction and Construction Learning' whose purpose is to force the network to understand the semantics of each region. For destruction, a region confusion mechanism (RCM) forces the classification network to learn from discriminative regions. For construction, the region alignment network restores the original region layout by modeling the semantic correlation among regions. A similar idea has been pursued in [38], where they propose a progressive training strategy to encourage the network to learn features at different granularities (using a random jigsaw patch generator) and afterwards fuse them together. Some other works introduce an attention mechanism. For instance, [227] proposes a novel part learning approach by a multi-attention convolutional neural network (MA-CNN) without bounding box/part annotations. MA-CNN jointly learns part proposals (defined as multiple attention areas with strong discrimination ability) and the feature representations on each part. Similar approaches have been reported in [117, 168]. In [36], they propose a network which learns sparse attention from class peak responses (which usually corresponds to informative object parts) and implements spatial and semantic sampling. Finally, in [77], the authors present an attention convolutional binary neural tree in a weakly-supervised approach. Different root-to-leaf paths in the tree network focus on different discriminative regions using the attention transformer inserted into the convolutional operations along edges of the tree. The final decision is produced as the summation of the predictions from the leaf nodes.

In another direction, some end-to-end frameworks aim to enhance the intermediate representation learning capability of a CNN by encoding higher-order statistics. For instance in [51] they capture the second-order information by taking the outer-product over the network output and itself. Other approaches focuses on reducing the high feature dimensionality [87] or extracting higher order information with kernelized modules [24]. In [192], they learn a bank of convolutional filters that capture class-specific discriminative patches without extra part or bounding box annotations. The advantage of this approach is that the network focuses on classification only and avoids the trade-off between recognition and localization.

Regardless, most fine-grained approaches use the object ground truth bounding box at test time, achieving a significantly lower performance when this information is not available. Moreover, automatically discovering discriminative parts might require large amounts of training images.

A summary of popular fine-grained image datasets is provided in Table 2.1. In this thesis, we will use the Oxford Flowers, CUB200, Stanford Dog and Stanford Cars fine-grained datasets.

3 Saliency for Fine-grained Object Recognition in Domains with Scarce Training Data¹

3.1 Introduction

Fine-grained object recognition focuses on the classification of subclasses belonging to the same category. Examples of fine-grained datasets include natural categories such as flowers [133], birds [199], dogs [85] and man-made categories such as cars [90]. The problem of fine-grained object classification is difficult because the differences between subclasses are often subtle and expert labelers, with knowledge of the discriminating attributes, are needed for the collection of datasets. Therefore the collection of large datasets is expensive and the development of algorithms that only require few labeled examples is of special interest to the field.

Computational saliency estimation aims to identify to what extent regions or objects stand out with respect to their surroundings to human observers. Saliency methods can be divided into methods that aim to identify the salient object (or objects) and methods that aim to produce a saliency map that is in accordance with measurements of human eye-movements on the same image. Itti et al. [76] proposed one of the first computational saliency methods based on combining the saliency cues for color, orientation and luminance. Many works followed proposing a large variety of hand-crafted features for saliency [11, 146]. Similar as other fields in computer vision, computational saliency estimation has moved in recent years

¹This chapter is based on a publication in Pattern Recognition 2019 [46].

from hand-designed features to end-to-end learned deep features [106].

Saliency detection in human vision plays a role in the efficient extraction of information by placing the attention on those regions in the image that are most informative. However, the vast majority of saliency methods are not evaluated on their efficiency to improve object recognition but instead are evaluated on the task of how accurate their generated saliency masks are. Given that saliency is only an intermediate step of the visual pipeline, evaluating the efficiency of saliency in terms of an improvement of the final task - here we consider fine-grained recognition - could be considered a more valuable evaluation. Therefore, in this chapter we aim to evaluate the usefulness of saliency by directly evaluating its improvement on image classification.

Previous works have found that the incorporation of attention mechanisms in neural networks could be beneficial. This theory was subsequently extended to captioning methods where the attention highlights the part of the image that is currently being described by words. Similar to these methods we will incorporate a saliency model, which modulates the normal forward pipeline similarly as an attention model would, but now within the context of fine-grained image classification. Contrarily to these attention methods, we use a saliency network that is pretrained on the task of saliency estimation. Especially, we are interested in demonstrating its effectiveness in the case of scarce training data, a scenario where attending to the relevant information from the image can significantly reduce the danger of overfitting. The main underlying idea is that using saliency as an attention mechanism can help backpropagation to focus on the relevant image information; something which is especially important when only few training examples are available.

In this chapter, we investigate to what extent saliency estimation can be exploited to improve the training of an object recognition model when scarce training data is available. For that purpose we design an image classification deep neural network that incorporates saliency information as input. This network processes the saliency map through a dedicated network branch and uses the resulting features to modulate the standard bottom-up visual features from the original image input. The main aim of the proposed method is to enable the effective training of a fine-grained recognition model with limited training samples and to improve the performance on the task, thereby alleviating the need to annotate a large dataset. We evaluate our method on different datasets and under different settings, achieving considerable performance improvements when leveraging saliency data, especially when training data is scarce.

The contributions of this chapter are as follow:

3.2. Saliency Modulation for Scarce Data Object Classification

- We investigated the role of saliency on improving the classification accuracy when the training data is scarce..
- We considered adding a saliency branch to an existing CNN architecture (AlexNet, ResNet-50 and ResNet-152).
- We validated our approach on the fine-grained object recognition problem.
- Experimental results confirmed that our approach is useful for the case when the available training data is scarce.
- Our experiments show that there exists a clear correlation (Pearson coefficient) between the performance of saliency methods on standard saliency benchmarks and the performance gain that is obtained when incorporating them in a object recognition pipeline

3.2 Saliency Modulation for Scarce Data Object Classification

Image classification results have improved much since the advent of deep convolutional neural networks [62, 93] due to the excellent visual representations learned by these models. Given the great number of parameters of these networks, we require large datasets of labeled data to effectively train them. For example the popular ImageNet dataset has over 1M labeled images [158]. Once learned, these strong image representations can be transferred to other related tasks by a process called finetuning. This process allows to use deep learning on tasks for which significantly less labeled data is available. In some cases, however, the available data for the target task is so scarce that is still insufficient to finetune large networks and obtain satisfactory results.

Saliency is an attentional mechanism which allows humans to focus their limited resources to the most relevant information in the image. Since processing resources are limited, the data is processed in a serial manner, prioritizing those parts that are expected to have high information content. In this chapter, we investigate another potential application of saliency, namely its function to facilitate the fast learning of new objects in the context of deep neural networks. Especially when only a few training examples are available, focusing on the relevant parts of the image could significantly improve the speed of learning, understanding speed as the number of example images required to learn a new class. Therefore, we seek

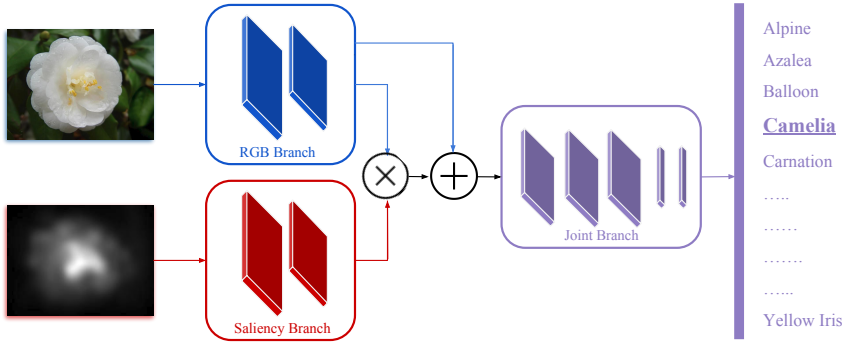


Figure 3.1: Overview of our fine-grained recognition model using saliency information. We process the two inputs, RGB and Saliency map, through two convolutional layers and then fuse the resulting features with a modulation layer. We then continue processing the fused features with three more convolutional layers and three fully connected layers, ending with the final classification layer.

to incorporate saliency estimation into an image classification pipeline, with the aim to decrease the data requirements for learning object categories.

Figure 3.1 provides an overview of the proposed network architecture. Our network contains two branches: one to process the RGB images and one to process their corresponding saliency images, which are pre-computed and given as input. They are combined with a *modulation layer* (\times symbol) and further processed by several shared layers of the joint branch to finally end on a classification layer. Note how the RGB branch followed by the joint branch correspond to a standard image classification network. The novelty of our architecture is the introduction of the saliency branch, which transforms the saliency image to the *modulation image*. This modulation image is then used to modulate the features of the RGB branch, putting more emphasis on those features that are considered important for the fine-grained recognition task. In the following sections we provide the details of the network architecture, the functioning of the modulation layer, and the saliency methods used.

We explain our model using AlexNet [93] as base classification network, but the theory could be applied to most convolutional neural network architectures. We also consider ResNet-50 and ResNet-152 [62] as base networks in our experiments (sec. 3.3.2).

3.2.1 Combining RGB with Saliency for Image Classification

Consider a saliency map $s(x, y)$ where x and y are the spatial coordinates. We will assume that saliency maps are of the same size as the original image $I(x, y, z)$, where $z = \{1, 2, 3\}$ indicate the three color channels of the image. A straightforward way to incorporate the saliency into the image classification network is by concatenating the image and the saliency map into an image with four channels such that $I(x, y, 4) = s(x, y)$. This strategy has been previously used by Murabito et al. [129] in a classification pipeline that combines two CNN networks: one to compute top-down saliency maps from an RGB image, and a second network that appends the generated saliency map to the RGB image channels to perform image classification.

In this case, the classification network only needs to train from scratch the weights of the first layer, the following layers can be initialized with a pretrained network. We call this approach *early fusion* of saliency and image content.

In this chapter we propose *delayed fusion* of saliency and image content, where we use the saliency map to modulate the features of an intermediate network layer. Consider the output of the i^{th} layer of the network, l^i , with dimension $w_i \times h_i \times z_i$. Then we define the modulation with a function $\hat{s}(x, y)$ as

$$\hat{l}^i(x, y, z) = l^i(x, y, z) \cdot \hat{s}(x, y), \quad (3.1)$$

yielding the saliency-modulated layer \hat{l}^i . Here the modulation image \hat{s} is the output of the saliency branch, which takes s as input (as depicted in Figure 3.1). Note that we consider a single saliency map \hat{s} that is independent of the number of feature maps. To ensure that \hat{s} has the same spatial dimensions as l^i , we use a similar architecture for both the saliency branch and the RGB branch. Concretely, the main difference resides in the size of the channel dimension: the saliency branch takes an intensity image as input (instead of a 3-channel RGB image) and outputs a scalar modulation image of $w_i \times h_i \times 1$ (instead of a $w_i \times h_i \times c_i$ feature map). Moreover, we use a sigmoid activation function at the end of the saliency branch, as opposed to the ReLU non-linearity of the RGB branch. This ensures that $0 \leq \hat{s}(x, y) \leq 1$ and thus provides a suitable range for feature modulation.

In the original architecture, max pooling is performed right after the second convolutional layer. In our model, we postpone this max pooling to after the features from both branches are fused, i.e. we perform max pooling on the saliency-modulated layer \hat{l}^i . The reasoning behind this choice is to leverage the greater modulation potential of higher resolution saliency features. We experimentally show (sec. 3.3.2) that this results in a small performance boost.

In addition to the formulation in Eq. (3.1) we also introduce a skip connection from the RGB branch to the beginning of the joint branch, defined as

$$\hat{l}^i(x, y, z) = l^i(x, y, z) \cdot (\hat{s}(x, y) + 1). \quad (3.2)$$

This skip connection is depicted in Figure 4.1 (+ symbol). It prevents the modulation layer from completely ignoring the features from the RGB branch. This is inspired by a previous work [211] that found this approach beneficial when using attention for network compression. We confirm the usefulness of the skip connection in the experiments section, sec. 3.3.2.

We train our architecture in an end-to-end manner. The backpropagated gradient from the modulation layer into the image classification branch is equal to

$$\frac{\partial L}{\partial l^i} = \frac{\partial L}{\partial \hat{l}^i} \cdot (s + 1), \quad (3.3)$$

where L is the loss function of the network. This shows that the saliency map not only modulates the forward pass (see Eq. (3.2)), but it also modulates the backward pass in exactly the same manner; in both cases putting more weight on the features that are on locations with high saliency, and putting less weight on the irrelevant features in the background on which the network could potentially overfit.

3.2.2 Training the Saliency Branch

The aim of the saliency branch is to process the saliency map $s(x, y)$ into effective modulation features $\hat{s}(x, y)$ that increase the classification performance when training with scarce data. The main intuition is that the saliency features \hat{s} will focus the backpropagated gradient to the relevant image features, thereby reducing the required data necessary to train the network. The additional saliency branch necessary to compute $\hat{s}(x, y)$ has its own set of parameters and could, in principle, increase the possibility of overfitting. We therefore consider two different scenarios to initialize this branch. In both cases, we start with an equivalent architecture to the one depicted in Figure 3.1 but without the saliency branch. We pretrain this network for image classification on ImageNet [158]. Then, we add the saliency branch and apply either of the following options:

- *Initialization from scratch*: the weights of the saliency branch are randomly initialized using the Xavier method.

3.2. Saliency Modulation for Scarce Data Object Classification

- *Initialization from pretrained*: the weights of the saliency branch are pretrained on an image classification network for which abundant training data is available. To do this, we first generate saliency images for the ImageNet validation dataset, which consists of 50K images (40K for training and 10K for validation) using the saliency method of choice. On this dataset we train our method, initializing the saliency branch from scratch. We now have a good pretrained model for the saliency branch too. Finally, we use this pretrained network (using both the saliency and RGB branch) to initialize all the weights of our network except the top classification layer.

3.2.3 Saliency input

The input to the saliency branch is a saliency map. Among the many saliency methods that provide satisfactory results [20], we perform most of our experiments using two of the top performing methods:

- iSEEL [174] leverages the inter-image similarities to train an ensemble of extreme learners. The predicted saliency of the input image is then calculated as the ensemble’s mean saliency value. Their approach is based on two aspects: (i) the contextual information of the scene and (ii) the influence of scene memorability (in terms of eye movement patterns by resemblance with past experiences). We use MATLAB code released by the authors.
- SALICON [72] exploits the power of high-level semantics encoded in a CNN pretrained on ImageNet. Their approach represents a breakthrough in saliency prediction, by reducing the semantic gap between the computational model and the human perception. Their method has two key elements: (i) an objective function based on saliency evaluation metrics and (ii) integration of information at different image scales. We use the open source implementation provided by [175].

Besides these two methods, we also perform experiments with three other approaches for a more comprehensive comparison.

- Itti and Koch [76]: First, we consider the classical saliency model of Itti et al. Several activation maps, corresponding to multiscale image features (color, intensity and orientations) are generated from the visual input and combined into a single topographical saliency map. A neural network is used to select

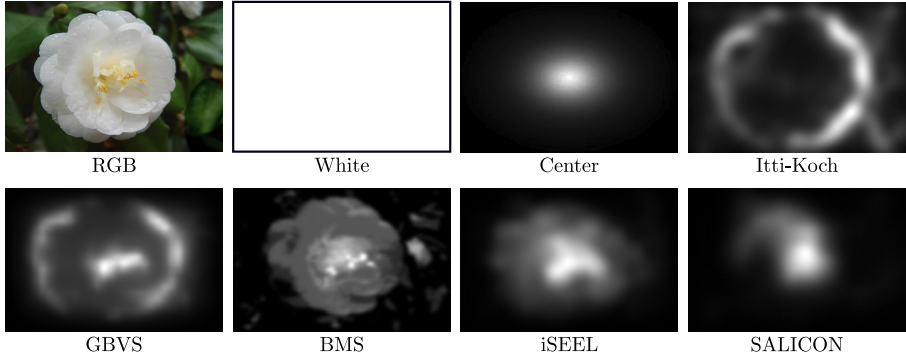


Figure 3.2: Saliency images generated with the different saliency estimation approaches considered, as well as the two baseline saliency maps evaluated, *White* and *Center*. We also include the original RGB image for reference.

the most salient locations in order of decreasing magnitude, which could be subsequently analyzed by more complex, higher cognitive level processes.

- GBVS [60]: The Graph-based Visual Saliency (GBVS) is also a biologically-plausible bottom-up model following the approach proposed earlier by Itti et al., but improving the performance of the generation of activation maps and the normalization/combination step. They used the Markovian formalism to describe the dissimilarity and concentration of salient locations of the image seen as a graph.
- BMS [217]: Boolean Map based Saliency (BMS) approach computes saliency by analyzing the topological structure of the Boolean maps. These maps are generated by randomly thresholding the color channels. As topological element they choose ‘surroundedness’ because it better characterizes the image/background segregation.

Figure 3.2 depicts the estimated saliency maps for an example image using the five different saliency methods presented above. In addition to these methods, we consider two additional saliency map baselines. *White* regards all image pixels as equally salient, and thus the saliency maps are uniformly white. On the other hand, *Center* emulates a center prior by representing saliency as a centered 2-dimensional Gaussian distribution. These two baselines allow us to determine whether our model is actually leveraging the saliency information contained in the maps, or it is simply adding a general image bias that is beneficial for recognition (e.g. center

bias). We are especially interested in assessing whether saliency methods that obtain higher performance on saliency benchmarks also yield better performance when incorporated into our saliency pipeline.

3.3 Experiments

3.3.1 Experimental Setup

Datasets. We have performed the evaluation of our approach on four standard datasets used for fine-grained classification

- *Flowers*: Oxford Flower 102 dataset [133] consists of 8189 images of flowers grouped in 102 classes. Each class contains between 40 and 258 images.
- *Birds*: is a dataset consisting of 11,788 images of bird species divided in 200 categories [199]. Each image is annotated with its bounding box and the image coordinates of 15 keypoints. However, in our experiments we used the whole image.
- *Cars*: the dataset in [90] contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been separated roughly in a 50-50 split.
- *Dogs*: Stanford Dogs [85] consists of 20,580 images of different breeds of dogs from around the world grouped in 120 categories. Since some of these images appear also in Imagenet, in our experiments we have discarded the repeated ones.

Networks. Our base network is AlexNet [93], which consists of five convolutional layers followed by three fully connected layers. We used the pretrained network on ImageNet [158] and fine-tuned it for fine-grained recognition on each dataset for 70 epochs with a learning rate of 0.01 and a weight decay of 0.003. The top classification layer is randomly initialized using Xavier. We have attached a saliency branch to this network as shown in Figure 3.1.

For some experiments we have also used the ResNet-50 and ResNet-152 [62], consisting of 50 and 152 convolutional layers, respectively, organized in 5 residual blocks. The structure of the saliency branch has been kept the same as in Figure 3.1,

i.e. consisting of two convolutional layers and having a ReLu activation function after the first one and a sigmoid function after the second.

Evaluation protocol. For all the above datasets, we randomly select and fix 5 images for test, 5 for validation, and keep the rest for training. We do this for each class in the dataset independently. In order to investigate different data scarcity levels, we train each model with subsets of k training images for $k \in \{1, 2, 3, 5, 10, 15, 20, 25, 30, K\}$, where K is the total number of training images for the class, which does not include the 10 held out images for validation and test. Contrarily to current few-shot approaches, this setting grants us a more complete disclosure of the results of our model under multiple limited-data scenarios. We use accuracy in terms of percentage of correctly classified samples as evaluation measure. We train and test each model five times with different random initializations, and show the average performance for the five runs.

3.3.2 Experimental Results

In the experimental section we evaluate the best strategies for fusing the saliency and RGB branches, compare several network architectures, evaluate various saliency methods as input to the saliency branch, and compare our results with state of the art on standard benchmark datasets for fine-grained object recognition.

Optimal architecture. In order to justify the design choices in our model, we present here multiple architectural variations to integrate saliency information into a neural network. We call *Baseline-RGB* to the original network model, which only contains the RGB branch and thus does not use any saliency information. We test an *Early fusion* model in which the saliency image is directly concatenated to the RGB input

We consider several variants of our model in which delayed fusion is performed at different network levels, indicated as Fusion L1 for fusion after layer 1 (similarly for Fusion L2, L3, L4, and L5). In all cases, we use a two-layer saliency branch, indicated by S2. Moreover, we evaluate whether performing the fusion after the pooling layer is a better option than doing it before. Finally, we include a model without the skip connection from the RGB branch to the joint branch (No SC).

We evaluate all models on *Flowers* [133] with AlexNet [93] and using iSEEL [174] as the saliency method of choice. Table 3.1 shows the results for different number of training images. First, we observe how the performance of all methods steadily

3.3. Experiments

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	88.0	68.5
Early Fusion	19.3	25.7	30.1	40.8	60.9	69.2	75.3	79.9	82.4	83.7	56.7
Fusion L1-S2	33.3	47.9	54.3	65.1	71.9	76.3	82.1	85.9	87.9	90.7	69.5
Fusion L2-S2	34.7	49.3	55.2	65.2	72.7	76.7	83.9	86.5	89.1	91.3	70.5
Fusion L3-S2	32.9	46.7	54.1	64.9	71.7	74.4	82.3	85.1	87.3	89.1	68.9
Fusion L4-S2	32.5	48.9	54.0	65.1	71.7	73.5	81.0	84.9	87.2	88.8	68.2
Fusion L5-S2	32.5	48.9	54.0	63.3	71.1	73.3	81.0	84.3	87.2	88.7	68.4
Fusion L2-S2 + After pool	34.3	49.1	55.5	66.0	72.1	77.5	83.6	85.6	88.9	90.2	70.2
Fusion L2-S2 + No SC	33.9	48.1	55.1	65.1	71.1	77.6	82.4	86.3	88.1	90.9	69.9

Table 3.1: Results for the baseline model and different variations of our architecture incorporating saliency information. The results correspond to the classification accuracy on the Flowers dataset [133] with AlexNet [93]. Each column indicates the number of training images used, and the rightmost column shows the average

grows when increasing the number of training images. In general, incorporating saliency information helps when fused within the network, but damages the accuracy if concatenated to the input image. We attribute this to the need to learn a low-level filter from scratch, which in turn affects the feature representation at higher levels. Performing the fusion immediately after the second convolutional layer seems to be the best option. Fusing before or after the pooling layer leads to similar results, the advantage of fusing higher resolution saliency features gives only a marginal boost. Finally, the skip connection from the RGB branch to the joint branch is also beneficial.

We have also explored different architectures for the saliency branch. We first assess whether an additional convolutional layer in the saliency branch leads to better performance. Table 3.2 presents the comparison between a two-layer saliency branch (S2) and a three-layer version (S3). For completeness, we explore merging after the second layer of the RGB branch (L2) as in previous experiments, and merging after the third layer (L3). We observe how an extra layer does not further improve the model’s performance. Alternatively, we investigate whether having fewer parameters in the saliency branch achieves higher results. We evaluate with 75% and 50% fewer parameters by reducing the number of output channels in the first layer. Table 3.3 shows how reducing the number of parameters in the saliency branch slightly reduces the final performance.

Pretraining the saliency branch on ImageNet. As described in section 3.2, we consider two alternative ways of initializing the saliency branch: from scratch and pretrained on ImageNet [158]. In this section, we compare these two approaches

Chapter 3. Saliency for Fine-grained Object Recognition

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	88.0	68.5
Fusion L2-S2	34.7	49.3	55.2	65.2	72.7	76.7	83.9	86.5	89.1	91.3	70.5
Fusion L3-S2	32.9	46.7	54.1	64.9	71.7	74.4	82.3	85.1	87.3	89.1	68.9
Fusion L2-S3	34.5	48.2	55.9	65.0	72.8	76.1	83.0	86.5	89.0	91.0	70.2
Fusion L3-S3	33.1	49.3	54.2	65.1	72.1	74.9	82.9	85.3	88.0	89.0	69.4

Table 3.2: Results on Flowers [133] with AlexNet [93] using two (S2) or three (S3) convolutional layers for the saliency branch.

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	88.0	68.5
Fusion L2-S2 (100%)	34.7	49.3	55.2	65.2	72.7	76.7	83.9	86.5	89.1	91.3	70.5
Fusion L2-S2 (75%)	34.7	49.0	55.3	65.1	72.0	77.0	83.3	85.9	88.3	89.1	70.0
Fusion L2-S2 (50%)	34.7	49.1	55.9	65.1	71.8	77.1	83.5	86.2	88.0	89.0	70.0

Table 3.3: Results on Flowers [133] with AlexNet [93] when reducing the number of parameters of the saliency branch.

with respect to the Baseline-RGB. The experiments are performed on *Flowers* dataset (see Figure 3.3a) and represent the classification accuracy versus the number of training samples. Adding a saliency branch initialized from scratch already outperforms the baseline using only RGB (see also Tab. 3.1), and pretraining this branch with ImageNet further increases the performance in a systematic and substantial manner. Our method with pretraining is especially advantageous in the scarce-data domain (i.e. < 20 images per class). For example, we obtain a better performance than the baseline using half the data, 10 images/class vs. 20 images/class, respectively. Furthermore, in the very low-range of number of samples we obtain similar performance with only *one third* of the samples (3 images/class vs. 10 images/class). Finally, our saliency branch is still beneficial even when using all available training samples. In fact, our method trained with a limited number of samples (around 25 per class) already surpasses the final performance of baseline using all samples.

Figure 3.4 shows some qualitative results for the case when the pretrained version of our approach predicts the correct label, meanwhile the Baseline-RGB fails. Alternatively, Figure 3.5 depicts the opposite case: the Baseline-RGB predicts the correct label of the test images, meanwhile the pretrained version of our approach fails. In both cases, the saliency images have been generated using the iSEEL method. A possible explanation for the failures in this latter case could be that the saliency images are not able to capture the relevant region of the image for fine-grained discrimination. Thus, the salience-modulated layer focuses on the

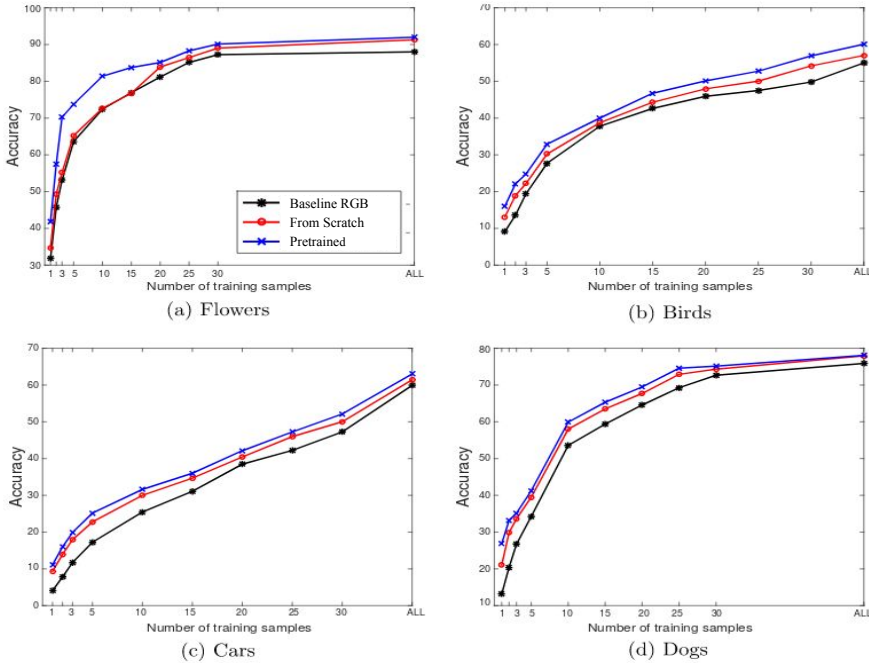


Figure 3.3: Experiments on four datasets using iSEEL [174] to generate the saliency maps. *Baseline-RGB* is compared against two different ways to initialize the saliency branch of our model: from scratch and pretrained on ImageNet [158].

wrong features for the task.

Different datasets.

Besides *Flowers* dataset, we validate our approach on three other datasets: *Birds*, *Cars* and *Dogs* (see figures 3.3b, c, and d, respectively). We follow the same experimental protocol as in the *Flowers* case. We can see how most trends observed in *Flowers* also apply to these datasets. For example, incorporating saliency information improves the classification accuracy, especially when data is scarce. Moreover, pretraining the saliency branch is beneficial for our method and leads to a further performance boost. Even when using all available samples, our method outperforms the baseline model. Therefore, we can claim that our approach successfully

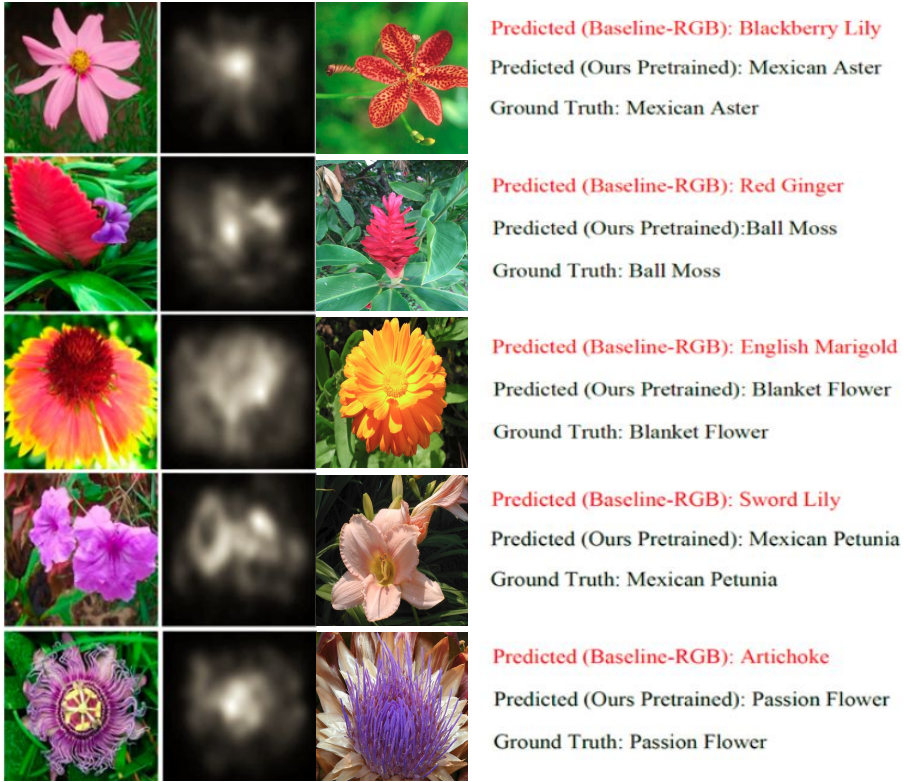


Figure 3.4: Some success examples on Flowers [133]: when the prediction done by Baseline-RGB fails to infer the right label for some test images, but the prediction by our approach is correct. From left to right: input image, saliency images generated with iSEEL [174], example image of the class with which the input image was wrongly predicted.

generalizes to other fine-grained datasets.

Confirmation of intuition. Our method is based on the idea that adding a saliency branch helps the network to focus on the relevant image regions during the training. To verify that this is actually happening we propose the following experiment: we measure if the percentage of backpropagated gradient magnitude which passes through the relevant image regions is increased by our proposed network architec-

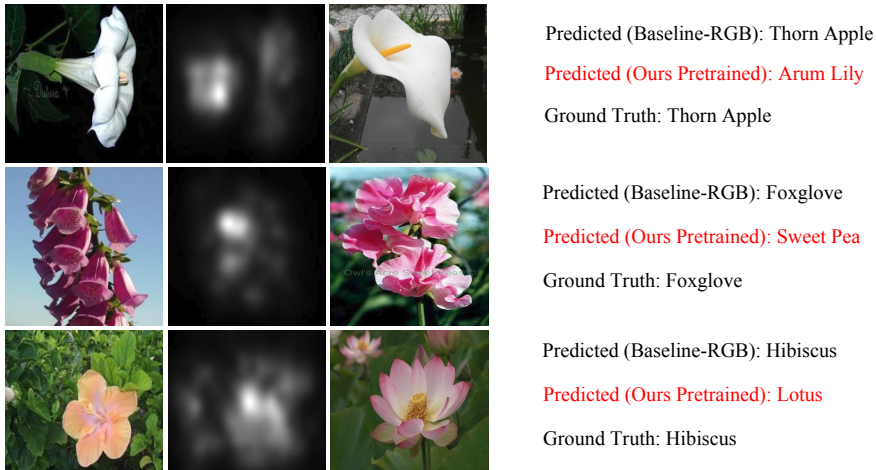


Figure 3.5: Some failure examples on Flowers [133]: when the prediction done by our method fails to infer the right label for some test images, but the prediction by Baseline-RGB is correct. From left to right: input image, saliency images generated with iSEEL [174], example image of the class with which the input image was wrongly predicted.

ture. We perform this experiment on the Birds dataset for which we have access to bounding box information of the birds (defining the relevant region). We measure the percentage of backpropagated gradient energy which is in the bounding box of the bird (this is computed by dividing the gradient magnitude in the bounding box by the gradient energy in the whole image). We measure this just before the third convolutional layer for AlexNet (which is just before the joint branch in Figure 3.1), and we measure this for both the network with and without saliency branch.

The results are presented in Figure 3.6. The results show that the percentage of backpropagated gradient that passes through the relevant image regions is higher for our approach. As expected it is even higher for the network with the pretrained saliency branch. However the gap with the network trained from scratch diminishes with the number of epochs. The fact that more backpropagated gradient energy goes through the relevant image regions may explain why our method obtains better results than the standard baseline method.

Different saliency methods. Table 3.4 presents results on the *Flowers* using our full

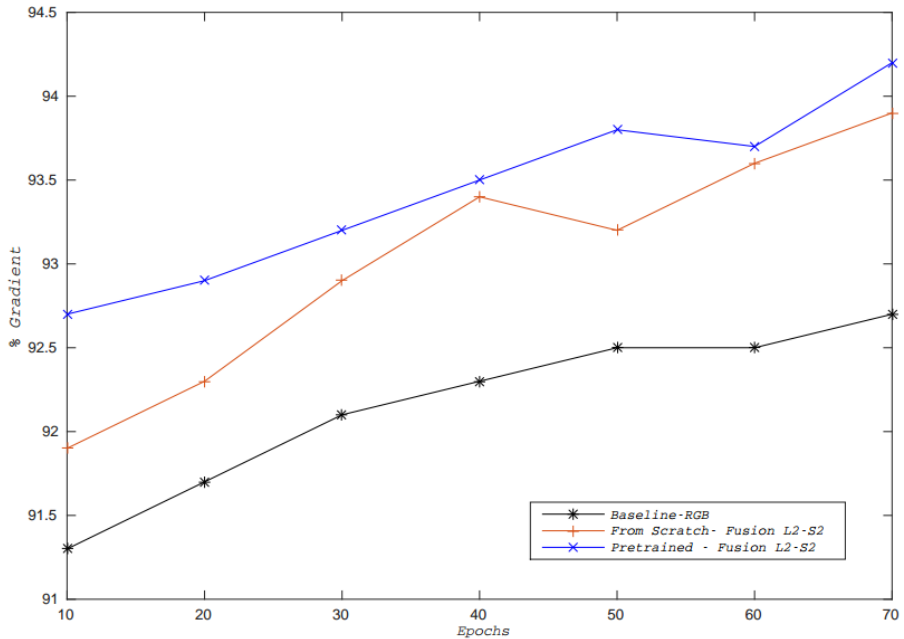


Figure 3.6: Average percentage of the total backpropagated gradient energy per epoch that is inside the bird bounding box. The graph shows that for our approach significantly more backpropagated gradient is on the relevant image region (for both the version trained from scratch and the version with pretrained saliency branch).

AlexNet model combined with the different input saliency maps. We can observe how, instead of helping, the two saliency baselines are actually hurting the method performance with respect to the Baseline-RGB. We hypothesize that this is due to the noise introduced in the network’s internal representation when the input saliency map is independent of the input image. On the other hand, all the saliency estimation methods increase the method performance, especially in the scarce-data range (i.e. < 10 images). Moreover, better saliency methods (e.g. iSEEL and SALICON) result in higher accuracies. In order to experimentally confirm this observation, we show in Figure 3.7 the accuracy of our image classification model as a function of the saliency estimation performance of the corresponding method. We measure saliency estimation performance in terms of Normalized Scanpath

Saliency (NSS), which is the official measure currently used by the popular MIT saliency benchmark [20] to sort all the participating methods. There is indeed a clear linear correlation, supported quantitatively by a Pearson product-moment correlation coefficient of 0.95. Therefore, we conclude that our model is agnostic to the saliency method employed. More importantly, it shows that better saliency methods (evaluated based on saliency estimation) actually lead to better image classification performance once integrated into an object recognition pipeline. This observation can be a motivation for saliency research: it not only leads to better saliency estimation but indirectly also contributes to improved object recognition.

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	88.0	68.5
Baseline-White	23.1	29.7	37.2	55.1	66.9	73	82.5	84.8	86.6	87.9	62.7
Baseline-Center	24.3	30.3	39.2	55.7	68.3	74.1	82.7	84.5	86.8	87.8	63.4
Itti-Koch [76]	32.8	46.8	53.9	64.0	72.9	77.1	82.9	85.4	87.1	88.3	69.1
GBVS [60]	33.3	46.9	54.0	64.1	73.0	77.3	83.1	85.7	87.5	88.8	69.4
BMS [217]	34.2	47.3	54.9	64.8	73.3	77.8	83.4	86.1	88.1	90.1	70.0
iSEEL [174]	34.7	49.3	55.2	65.2	72.7	76.7	83.9	86.5	89.1	91.3	70.5
SALICON [80]	37.6	51.9	57.1	68.5	75.2	79.7	84.9	88.2	91.2	92.4	72.7

Table 3.4: Comparison of different saliency methods regarding the effect on our model. The results correspond to the classification accuracy on the Flowers dataset [133] when using our full model with AlexNet [93] as base network. Each column indicates the number of training images used, and the rightmost column shows the average.

Different base networks. In order to evaluate the generality of our approach across different base networks, we have considered ResNet-50 and ResNet-152 as alternatives to AlexNet. We have tested several possible fusion architectures (Tables 3.5 and 3.6), but the optimal performance has been obtained when the fusion between the RGB and saliency branches takes place after the fourth residual block, with a two-layer saliency branch (Block4-S2). Results in Table 3.7 show the classification accuracy achieved on *Flowers* when using ResNet-50 and ResNet-152 with SALICON saliency maps. Furthermore, we compared our two initialization methods for the saliency branch (from scratch and pretrained on ImageNet) against the Baseline-RGB. Although under both initializations we obtained higher accuracy, the one that performs the best is the pretrained. These results confirm the trend already observed for AlexNet regarding the benefits of pretraining the saliency branch as shown in Figure 3.3.

Comparison with standard dataset splits. All previous experiments use a custom

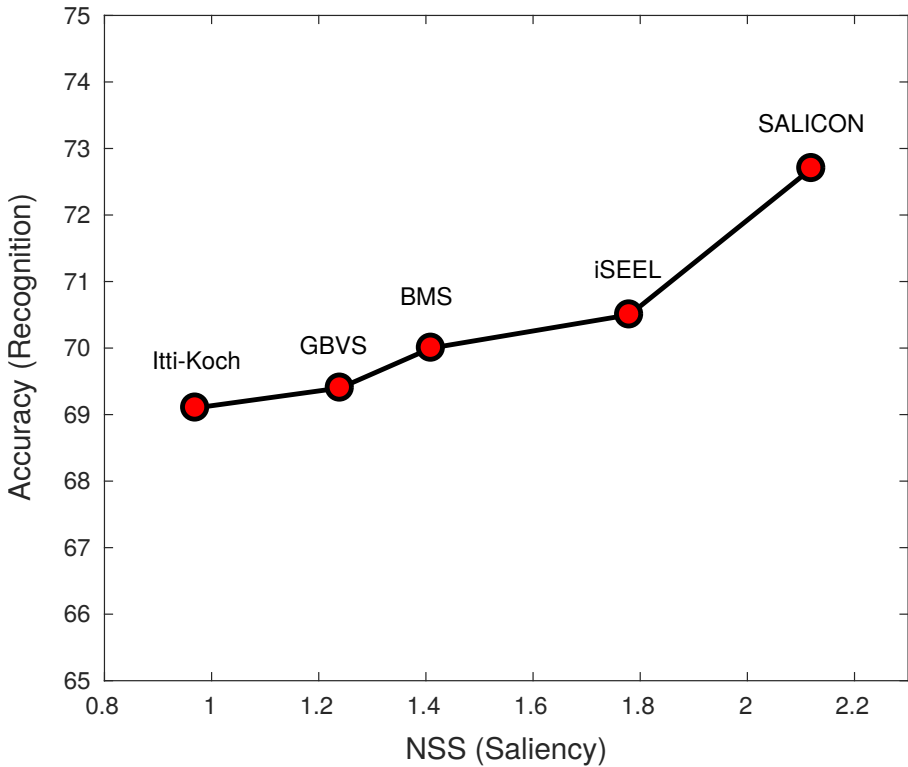


Figure 3.7: Correlation between the performance of the saliency method in terms of NSS and the fine-grained recognition accuracy of our method using the corresponding saliency model. Results with AlexNet [93] on Flowers [133].

data split consisting of a fixed test set of 5 images and a varying number of training images. In order to enable comparisons with published results by other methods, we perform here experiments using the standard data split of each dataset, employing the entirety of the corresponding given sets for training and evaluation. Table 3.8 presents results for our approach and several state of the art fine-grained recognition approaches for Flowers, Birds, and Cars datasets. We discard Dogs dataset due to the overlap with the ImageNet images already used for pretraining the network, as they can no longer be ignored when using the full sets. Our approach uses SALICON saliency and ResNet152 as base network, which is equivalent

3.3. Experiments

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	39.1	59.6	67.8	81.6	89.0	91.7	92.7	93.0	93.0	95.4	80.3
Block1-S2	38.0	59.2	68.0	80.7	88.8	91.0	91.9	92.0	92.1	94.8	79.6
Block2-S2	38.2	59.5	68.0	81.4	90.0	91.6	92.0	92.4	93.0	94.9	80.1
Block3-S2	39.3	62.9	68.5	83.0	90.0	92.1	93.5	94.9	93.4	95.9	81.4
Block4-S2	45.8	64.3	72.8	83.0	90.5	93.0	93.9	94.6	93.7	96.7	82.7
Block5-S2	38.2	57.9	65.9	80.8	87.1	90.9	91.1	91.2	91.9	92.6	78.8

Table 3.5: Results for the baseline model and different variations of our architecture incorporating saliency information in different blocks. The results correspond to the classification accuracy on the Flowers dataset [133] with ResNet-50 [62]. Each column indicates the number of training images used, and the rightmost column shows the average

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB	39.0	60.1	68.0	82.5	89.0	92.0	92.1	93.3	94.2	95.8	80.6
Block1-S2	39.0	59.9	68.0	82.1	88.6	91.9	92.2	93.0	94.2	95.1	80.4
Block2-S2	38.8	60.2	68.2	83.0	90.2	92.2	93.0	94.0	94.0	96.2	81.0
Block3-S2	43.0	63.7	68.9	83.1	90.2	92.1	93.1	94.3	96.1	96.3	82.1
Block4-S2	42.6	64.2	70.9	85.5	90.9	92.7	94.0	95.0	97.0	97.8	83.1
Block5-S2	39.0	58.0	65.8	80.3	87.1	90.8	91.5	92.0	92.3	92.7	79.0

Table 3.6: Results for the baseline model and different variations of our architecture incorporating saliency information in different blocks. The results correspond to the classification accuracy on the Flowers dataset [133] with ResNet-152 [62]. Each column indicates the number of training images used, and the rightmost column shows the average

to the networks used by the most recent works. Our method is competitive with specialized fine-grained approaches, despite the more sophisticated techniques included in those (e.g. part localization), some of which might be complementary to our saliency modulation. Moreover, our approach is especially beneficial in the scarce training data regime, whereas some of the state of the art methods may not work under these conditions.

Comparison with few-shot method. Our scarce-data approach is similar in spirit to the few-shot learning methods [128, 164, 186]. For this reason, we propose here a comparison with the state of the art method for few-shot classification, Prototypical networks [164]. In the standard few-shot protocol, the task is framed as N -way k -shot, i.e. provide each time a set of k labeled samples from each of N classes that

Chapter 3. Saliency for Fine-grained Object Recognition

Method	1	2	3	5	10	15	20	25	30	K	AVG
Baseline-RGB Resnet-50	39.1	59.6	67.8	81.6	89.0	91.7	92.7	93.0	93.0	95.4	80.3
Resnet-50 Block4-S2 From Scratch	45.8	64.1	71.8	83.0	90.5	93.0	93.9	94.6	93.7	96.7	82.7
Resnet-50 Block4-S2 Pretrained	47.1	65.2	72.9	83.8	91.3	93.9	94.6	95.4	94.7	97.4	83.6
Baseline-RGB Resnet-152	39.0	60.1	68.0	82.5	89.0	92.0	92.1	93.3	94.2	95.8	80.6
Resnet-152 Block4-S2 From Scratch	42.6	64.2	70.9	85.5	90.9	92.7	94.0	95.0	97.0	97.8	83.1
Resnet-152 Block4-S2 Pretrained	46.9	65.5	73.0	84.7	92.0	94.2	95.3	95.8	97.3	98.1	84.3

Table 3.7: Results on Flowers [133] using ResNet-50 and Resnet-152 [62] as base networks and SALICON [80] as saliency method.

Method	Flowers	Birds	Cars
Krause et al. [91]	-	82.0	92.6
RA-CNN [49]	-	85.3	92.5
Bilinear-CNN [109]	-	84.1	91.3
Compact Bilinear Pooling [51]	-	84.3	91.2
Low-rank Bilinear Pooling [88]	-	84.2	90.9
Cui et al. (with Imagenet) [32]	96.3	82.8	91.3
MA-CNN [227]	-	86.5	92.8
Ge-Yu [53]	90.3	-	-
DLA [210]	-	85.1	94.1
Ours (Resnet152 Block4-S2 From Scratch)	96.4	85.6	92.1
Ours (Resnet152 Block4-S2 Pretrained)	97.8	86.1	92.4

Table 3.8: Comparison with state of the art methods for domain-specific fine-grained recognition using the standard data splits of Flowers [133], Birds [199] and Cars [90]. Our approach uses ResNet-152 [62] as base network and SALICON [80] saliency maps.

have not previously been trained upon. The goal is then to classify a disjoint batch of unlabeled samples, known as ‘queries’, into one of these N classes. Therefore, some classes are used to train the few-shot method, while others are only used at test time. In our case, we do not require such split, as we can train and test the model in all classes simultaneously. Moreover, their test episodes are composed of only N classes at a time, where N is generally a small number (e.g. below 20). Contrarily, we follow a more general classification approach and test on all classes simultaneously, which is inherently more challenging as the classification probability increases.

We propose two different scenarios to compare our method to Prototypical networks on the task of Flower [133] classification. The first, *20-way 5-shot*, closely

Method	20-way 5-shot	102-way 5-shot
Prototypical networks [164]	53.8	26.2
Ours	81.0	73.8

Table 3.9: Results for few-shot classification on Flowers [133] when using our full model with AlexNet [93] as base network.

resembles the setting introduced by [186] and usually employed by few-shot approaches. We split the set of classes in train and test, selecting 20 random classes for the testing phase. Then, we run Prototypical networks for the 20-way 5-shot classification task, following similar settings to those used in the mini-ImageNet experiment of [164]. We train until convergence using 100 training episodes and test using 5 episodes, with 5 queries per episode both during training and testing. The second scenario, *102-way 5-shot*, is more similar to the conventional classification task, in which all classes are used for training and testing. We maintain the training settings for this case, but remove from the ‘shot’ set those queries used at test time. Table 3.9 presents the results of these experiments. Our method leads to substantially superior performance in both cases, but the difference is especially remarkable for the 102-way setting. This demonstrates the limitations of this type of few-shot approaches when scaling to many classes, even when they are trained with the same set of classes used for test.

3.4 Conclusion

In this chapter, we investigated the role of saliency in improving the classification accuracy of a CNN when the available training data is scarce. For that purpose we have considered adding a saliency branch to an existing CNN architecture, which is used to modulate the standard bottom-up visual features from the original input image. We have shown that the proposed approach leads to an improvement of the recognition accuracy with limited number of training data, when applied to the task of fine-grained object recognition.

Extensive evaluation has been performed on several datasets and under different settings, demonstrating the usefulness of saliency for fine-grained object recognition, especially for the case of scarce training data. In addition, our approach allows to compare saliency methods on the high-level task of fine-grained object recognition. Traditionally, saliency methods are evaluated on their ability

to generate saliency maps that indicate the relative relevance of regions for the human visual system. However, it remained unclear if these saliency methods would actually translate into improved high-level vision results for tasks such as object recognition. Our experiments show that there exists a clear correlation (Pearson product-moment correlation coefficient of 0.95) between the performance of saliency methods on standard saliency benchmarks and the performance gain that is obtained when incorporating them in a object recognition pipeline.

4 Saliency for Object Recognition, and Object Recognition for Saliency¹

4.1 Introduction

Fine-grained image recognition has as objective to recognize many sub-categories of a super-category. Examples of well-known fine-grained datasets are Flowers [133], Cars [90] and Birds [199]. The challenge of fine-grained image recognition is that the differences between classes are often very subtle, and only the detection of small highly localized features will correctly lead to the recognition of the specific bird or flower species. An additional challenge of fine-grained image recognition is the difficulty of data collection. The labelling of these datasets requires experts and subcategories can be very rare which further complicates the collection of data. Therefore, the ability to train high-quality image classification systems from few data is an important research topic in fine-grained object recognition.

Most of the state-of-the-art general object classification approaches [93, 188] have difficulties in the fine-grained recognition task, which is more challenging due to the fact that basic-level categories (e.g. different bird species or flowers) share similar shape and visual appearance. Early works have focused on localization and classification of semantic parts using either explicit annotation [108, 214, 221] or weakly labeling [50, 227]. The main disadvantage of these approach was that they

¹This chapter is based on a paper accepted as a Full Paper in VISAPP 2021 [47].

Chapter 4. Hallucinating Saliency Maps for Fine-Grained Image Classification for Limited Data Domains

required two different 'pipelines', for detection and classification, which made more complicated the joint optimization of the two subsystems. Therefore, more recent approaches are proposing end-to-end strategies with the focus on improving the feature representation from intermediate layers in a CNN through higher order statistics modeling [24, 192].

In the previous chapter we have discussed a fine-grained recognition system that only requires few labelled data. We will refer to this technique as saliency-modulated image classification (SMIC). This is especially beneficial when only few labelled data is available. The gradients which are backpropagated are concentrated on the regions which have high saliency. This prevents backpropagation of gradients of uninformative background parts of the image which could lead to overfitting to irrelevant details. A major drawback of this approach is that it requires an explicit saliency algorithm which needs to be trained on a saliency dataset.

In order to overcome the lack of sufficient data for a given modality, a common strategy is to introduce a 'hallucination' mechanism which emulates the effect of genuine data. For instance, in [68], they use this 'hallucination' strategy for RGB-D object detection. A hallucination network is trained to learn a complementary RGB image representation which is taught to mimic convolutional mid-level features from a depth network. At test time, images are processed jointly through the RGB and hallucination networks, demonstrating an improvement in detection performance. This strategy has been adopted also for the case of few-shot learning [61, 193, 215]. In this case, the hallucination network has been used to produce additional training samples used to train jointly with the original network (also called a meta-learner).

In this chapter, we address the major drawback of SMIC, by implementing a hallucination mechanism in order to remove the requirement for providing saliency images for training obtained using one of the existing algorithms [20]. In other words, we show that the explicit saliency branch which requires training on a saliency image dataset, can be replaced with a branch which is trained end-to-end for the task of image classification (for which no saliency dataset is required). We replace the saliency image with the input RGB image (see Figure 3.1). We then pre-train this network for the task of image classification using a subset from the ImageNet validation dataset. During this process, the saliency branch will learn to identify which regions are more discriminative. In a second phase, we initialize the weights of the saliency branch with these pre-trained weights. We then train the system end-to-end on the fine-grained dataset using only the RGB images. Results show that the saliency branch improves fine-grained recognition

significantly, especially for domains with few training images.

We briefly summarize below our main contributions in this chapter:

- we propose an approach which hallucinates saliency maps that are fused together with the RGB modality via a modulation process,
- our method does not require any saliency maps for training (like in these works [46, 129]) but instead is trained indirectly in an end-to-end fashion by training the network for image classification,
- our method improves classification accuracy on three fine-grained datasets, especially for domains with limited data.

4.2 Proposed Method

Several works have shown that having the saliency map of an image can be helpful for object recognition and fine-grained recognition in particular [129]. The idea is twofold: the saliency map can help focus the attention on the relevant parts of the image to improve the recognition, and it can help guide the training by focusing the backpropagation to the relevant image regions. In the previous chapter, we show that saliency-modulated image classification (SMIC) is especially efficient for training on datasets with few labeled data. The main drawback of these methods is that they require a trained saliency method. Here we show that this restriction can be removed and that we can hallucinate the saliency image from the RGB image. By training the network for image classification on the imageNet dataset we can obtain the saliency branch without human ground truth images.

4.2.1 Overview of the Method

The overview of our proposed network architecture is illustrated in Figure 4.1. Our network consists of two branches: one to extract the features from an RGB image, and the other one (saliency branch) to generate the saliency map from the same RGB image. Both branches are combined using a *modulation layer* (represented by the \times symbol) and are then processed by several shared layers of the joint branch which finally ends up with a classification layer. The RGB branch followed by the joint branch resembles a standard image classification network. The novelty of our architecture is the introduction of the saliency branch, which transforms the generated saliency image into a *modulation image*. This modulation image is used

Chapter 4. Hallucinating Saliency Maps for Fine-Grained Image Classification for Limited Data Domains

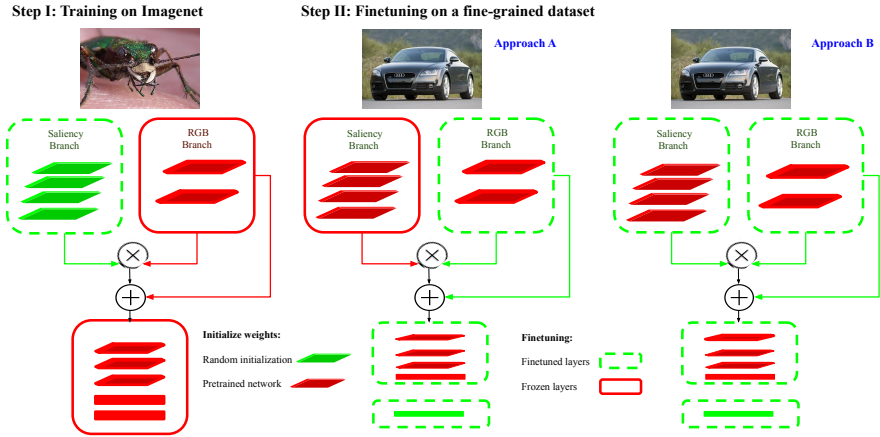


Figure 4.1: Overview of our method. We process an RGB input image through two branches: one branch extracts the RGB features and the other one is used to learn saliency maps. The resulting features are merged via a modulation layer, which continues with a few more convolutional layers and a classification layer. The network is trained in two steps.

to modulate the characteristics of the RGB branch, putting more emphasis on those characteristics that are considered important for the fine-grained recognition task. In the following sections we provide the details of the network architecture, the operation of the modulation layer, and finally, how our saliency map is generated. We explain our model using AlexNet [93] as the base classification network, but the theory could be extended to other convolutional neural network architectures. For instance, in the experimental results section, we also consider the ResNet-152 architecture [62].

4.2.2 Hallucination of saliency maps from RGB images

The function of the visual attention maps is to focus on the location of the characteristics necessary to identify the target classes, ignoring anything else that may be irrelevant to the classification task. Therefore, given an input RGB image, our saliency branch should be able to produce a map of the most salient image locations useful for classification purposes.

To achieve that, we apply a CNN-based saliency detector consisting of four convolutional layers (based on the AlexNet architecture)². The output from the last convolutional layer, i.e. one with 384 dimensional feature maps with a spatial resolution of 13×13 (for a 227×227 RGB input image), is further processed using a 1×1 convolution and then a function of activation ReLU. This is to calculate the saliency score for each "pixel" in the feature maps of the previous layer, and to produce a single channel map. Finally, to generate the input for the subsequent classification network, the 13×13 saliency maps are upsampled to 27×27 (which is the default input size of the next classification module) through bilinear interpolation. We justify the size of the output maps by claiming that saliency is a primitive mechanism, used by humans to direct attention to objects of interest, which is evoked by coarse visual stimuli. Therefore, our experiments (see section IV) show that 13×13 feature maps can encode the information needed to detect salient areas and drive a classifier with them.

4.2.3 Fusion of RGB and Saliency Branches

Consider an input image $I(x, y, z)$, where $z = \{1, 2, 3\}$ indicate the three color channels of the image. Also consider a saliency map $s(x, y)$. In Flores et al. [46], a network $h(I, s)$ was trained which performed image classification based on the input image I and the saliency map s . Here, we replace the saliency map (which was generated by a saliency algorithm) by a hallucinated saliency map $h(I, \hat{s}(I))$. The hallucinated saliency map \hat{s} is trained end-to-end and estimated from the same input image I without the need of any ground truth saliency data.

The combination of the hallucinated saliency map \hat{s} , which is the output of the saliency branch, and the RGB branch is done with modulation. Consider the output of the i^{th} layer of the network, l^i , with dimension $w_i \times h_i \times z_i$. Then we define the modulation as

$$\hat{l}^i(x, y, z) = l^i(x, y, z) \cdot \hat{s}(x, y), \quad (4.1)$$

resulting in the saliency-modulated layer \hat{l}^i . Note that a single hallucinated saliency map is used to modulate all i feature maps of \hat{l} .

In addition to the formula in Eq. (4.1) we also introduce a skip connection from

²We vary the number of convolutional layers in the experimental section and found four to be optimal.

the RGB branch to the beginning of the joint branch, defined as

$$\hat{l}^i(x, y, z) = l^i(x, y, z) \cdot (\hat{s}(x, y) + 1). \quad (4.2)$$

This skip connection is depicted in Figure 4.1 (+ symbol). It prevents the modulation layer from completely ignoring the features from the RGB branch.

We train our architecture in an end-to-end manner. The backpropagated gradient for the modulation layer into the image classification branch is equal defined as:

$$\frac{\partial L}{\partial l^i} = \frac{\partial L}{\partial \hat{l}^i} \cdot (\hat{s}(x, y) + 1), \quad (4.3)$$

where L is the loss function of the network. We can see that the saliency map modulates both the forward pass (see Eq. (4.2)) as well as the backward pass in the same manner; in both cases putting more weight on the features that are on locations with high saliency, and putting less weight on the irrelevant features. We show in the experiments that this helps the network train more efficiently, also on datasets with only few labeled samples. The modulation prevents the network from overfitting to the background.

4.2.4 Training on Imagenet and fine-tuning on a target dataset

As can be seen in Figure 4.1, the training of our approach is divided into two steps: first, training on Imagenet and second, fine-tuning on a target dataset.

Step 1: Training of saliency branch on Imagenet.

As explained above, the aim of the saliency branch is to hallucinate (generate) a saliency map directly from an RGB input image. This network is constructed by initializing the RGB branch with pretrained weights from Imagenet. The weights of the saliency branch are initialized randomly using the Xavier method (see Figure 4.1, left image). The network is then trained selectively, using the ImageNet validation set: we allow to train only the layers corresponding to the saliency branch (depicted by the surrounding dotted line) and freeze all the remaining layers (depicted through the continuous line boxes).

Step 2: Fine-tuning on a target dataset. In this step, we initialize the RGB branch with the weights pre-trained from Imagenet and the saliency branch with the corresponding pre-trained weights from Step 1. The weights of the top classification

layer are initialized randomly, using the Xavier method. Then, this network is then further fine-tuned on a target dataset, selectively. We distinguish two cases:

- Approach A: We freeze the layers of the saliency branch and we allow all the other layers in the network to be trained. This process is depicted by the continuous line surrounding the saliency branch and the dotted line for the rest (see the Figure 4.1, middle image).
- Approach B: We allow all layers to be trained. Since we consider training on datasets with only few labels this could result in overfitting, since it requires all the weights of the saliency branch to be learned (see the Figure 4.1, right image).

In the experiments we evaluate both approaches to training the network.

4.3 Experimental Results

4.3.1 System setup

Datasets. To evaluate our approach, we used three standard datasets used for fine-grained image classification:

- *Flowers*: Oxford Flower 102 dataset [133] has 8,189 images divided in 102 classes.
- *Birds*: CUB200 has 11,788 images of 200 different bird species [199].
- *Cars*: the CARS-196 dataset in [90] contains 16,185 images of 196 car classes.

Networks architectures. We evaluate our approach using two network architectures: Alexnet [93] and Resnet-152 [62]. In both cases, the weights were pretrained on Imagenet and then finetuned on each of the datasets mentioned above. The networks were trained for 70 epochs with a learning rate of 0.0001 and a weight decay of 0.005. The top classification layer was initialized from scratch using Xavier method [56].

Evaluation protocol. To validate our approach, we follow the same protocol as in [46]. For the image classification task, we train each model with subsets of k training images for $k \in \{1, 2, 3, 5, 10, 15, 20, 25, 30, K\}$, where k is the total number of training

images for the class. We keep 5 images per class for validation and 5 images per class for test. We report the performance in terms of accuracy, i.e. percentage of correctly classified samples. We show the results as an average over three runs.

4.3.2 Fine-grained Image Classification Results

Optimal depth and fusion saliency branch: We first evaluate the saliency branch with a variable number of convolutional layers. The results are presented in Figure 4.2 and we found that four convolutional layers led to a significant increase in performance, this can also be seen in more detail in Table 4.1, where we show the accuracy classification for Flowers-102, Birds and Cars, using AlexNet as base Network. Also, we look for the best RGB branch layer to perform the saliency branch and RGB branch merge. The results are presented in Figure 4.3, and in more detail in Table 4.2 for the aforementioned datasets.

It is found to be optimal to fuse the two branches before the Pool-2 layer for AlexNet³. Based on these experiments, we use four convolutional layers in the saliency branch and fuse before the second pool layer for the remainder of the experiments and for all datasets.

Evaluation on scarce data domain: As described in section III, we consider two alternative ways to train the saliency branch on the target dataset: keeping the saliency branch fixed (Approach A) or allowing it to finetune (Approach B). In this section, we compare these two approaches with respect to the Baseline-RGB and Baseline-RGB + scratch SAL (where Saliency branch is initialized from scratch without pretraining on Imagenet) and the work presented in the previous chapter, called SMIC. We do not compare to other fine-grained methods here, because they do not report results when only considering few labeled images. The experiments are performed on *Flowers*, *Cars* and *Birds* datasets and can be seen in Table 4.3. The average improvement of accuracy of our *Approach A* and *B* with respect the *Baseline-RGB* is 3.7% and 4.3%, respectively for the *Flowers* dataset; 3.9% and 4.3%, respectively for the *Cars* dataset; and 2.4% and 2.9%, respectively for the *Birds* dataset. Our *Approach B* is especially advantageous, if we compare it with the our previous SMIC approach, where we needed an additional algorithm to generate the salience map. It is therefore advantageous to also finetune the saliency branch on the target data even when we only have a few labeled images per class.

In Table 4.4, we show the same results but now for ResNet152. One can see

³In a similar study, we found that for Resnet-152 the optimal fusion is after the forth residual block.

4.3. Experimental Results

	#train images	1	2	3	5	10	15	20	25	30	K	AVG
Flowers	Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	87.8	68.3
	1 Conv Layer	32.0	46.3	53.9	64.4	72.9	76.7	81.4	85.3	89.1	87.9	69.0
	2 Conv Layer	33.8	47.7	54.7	66.5	73.5	77.0	82.0	86.1	90.3	89.1	70.1
	3 Conv Layer	35.1	48.8	56.3	68.1	74.9	77.9	82.7	87.0	91.0	91.2	71.3
	4 Conv Layer	37.3	51.7	57.2	68.7	75.6	78.7	83.8	88.4	91.7	92.5	72.6
Cars	Baseline-RGB	4.1	7.8	11.7	17.3	25.5	31.1	38.5	42.2	47.2	60.0	28.5
	1 Conv Layer	4.3	8.3	12.4	18.1	25.7	31.9	38.4	43.0	47.6	60.2	29.0
	2 Conv Layer	6.2	9.4	15.6	20.1	26.2	33.0	38.7	44.6	48.7	60.9	30.3
	3 Conv Layer	8.7	11.8	17.3	21.4	27.0	34.1	39.1	45.0	49.0	61.5	31.5
	4 Conv. Layer	9.8	15.1	18.4	22.9	28.8	35.1	39.9	45.8	49.7	62.9	32.8
Birds	Baseline-RGB	9.1	13.6	19.4	27.7	37.8	44.3	48.0	50.0	54.2	57.0	34.8
	1 Conv. Layer	9.9	14.3	20.0	27.9	38.0	44.0	47.7	48.9	52.9	57.1	36.1
	2 Conv. Layer	10.2	15.0	21.4	28.3	38.4	44.1	48.1	49.0	53.3	57.1	36.5
	3 Conv. Layer	11.8	16.9	22.1	29.0	38.9	44.5	48.4	49.3	53.9	57.4	37.2
	4 Conv. Layer	12.9	18.7	22.7	29.7	39.4	44.1	48.2	49.9	53.9	57.7	37.7

Table 4.1: Results on Flowers, Birds and Cars (results are the average over three runs), using AlexNet as base network. For the baseline model and different variations of layers on saliency branch of our architecture for saliency detection . The results correspond to the classification accuracy. Each column indicates the number of training images used, and the rightmost column shows the average.

that the results improve significantly, especially for *Cars* results improve a lot. The same general conclusions can be drawn : *Approach B* obtains better results than *Approach A* and the method obtains similar results as SMIC but without the need of a pretrained saliency network.

Comparison with other state-of-the-art approaches: In the past experiments, we used a custom data split consisting of a fixed subset of k training images. To compare our approach with other state-of-the-art methods, we followed the standard data split for training and evaluation of each dataset. Note that our main purpose is to evaluate on domains with little labeled data but we have included this results for comparison. This results are presented in Table 4.5. For the current comparison, we use our both approaches with ResNet-152 as base network, which is equivalent to the network architecture used by the most of the recent works. It can be appreciated that both our methods show similar performance with other fine-grained specialized approaches which often use more complex architectures including part-localization modules.

Qualitative results: Table 4.6 shows some qualitative results for the case when the pretrained version of our approach predicts the correct label, meanwhile the

Chapter 4. Hallucinating Saliency Maps for Fine-Grained Image Classification for Limited Data Domains

	#train images	1	2	3	5	10	15	20	25	30	K	AVG
Flowers	Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	87.8	68.3
	Merge Before Pool 2	37.3	51.7	57.2	68.7	75.6	78.7	83.8	88.4	91.7	92.5	72.6
	Merge After Pool 2	37.0	51.8	57.0	68.9	75.4	78.2	83.7	88.4	91.2	89.7	72.1
	Merge After Conv. 3	35.9	50.9	56.9	67.8	73.9	77.4	82.9	86.1	90.7	89.4	71.2
	Merge After Conv. 4	34.3	48.8	55.1	65.0	72.7	77.0	81.9	85.2	88.9	88.3	69.7
Cars	Baseline-RGB	4.1	7.8	11.7	17.3	25.5	31.1	38.5	42.2	47.2	60.0	28.5
	Merge Before Pool 2	9.8	15.1	18.4	22.9	28.8	35.1	39.9	45.8	49.7	62.9	32.8
	Merge After Pool 2	9.3	14.7	18.7	22.0	28.9	35.3	40.1	45.5	49.3	61.0	32.5
	Merge After Conv. 3	8.5	12.1	15.0	19.3	27.7	34.2	38.7	44.0	48.1	60.0	30.8
	Merge After Conv. 4	7.1	10.3	13.1	18.9	26.1	32.9	38.1	42.9	47.8	59.1	29.6
Birds	Baseline-RGB	9.1	13.6	19.4	27.7	37.8	44.3	48.0	50.0	54.2	57.0	34.8
	Merge Before Pool 2	12.9	18.7	22.7	29.7	39.4	44.1	48.2	49.9	53.9	57.7	37.7
	Merge After Pool 2	9.0	17.9	22.8	29.1	38.9	44.4	48.5	50.1	53.8	56.8	37.1
	Merge After Conv. 3	8.1	15.3	20.7	28.3	38.3	43.1	47.3	49.2	53.0	55.9	35.9
	Merge After Conv. 4	6.1	14.2	20.0	27.9	38.0	42.9	47.0	48.1	52.1	55.2	35.2

Table 4.2: Results on Flowers, Birds and Cars (results are the average over three runs), using AlexNet as base network. For the baseline model and different variations of our architecture incorporating the merge of our hallucinating saliency. The results correspond to the classification accuracy. Each column indicates the number of training images used, and the rightmost column shows the average.

Baseline-RGB fails. Alternatively, in Table 4.7 depicts the opposite case: the Baseline-RGB predicts the correct label of the test images, meanwhile the pretrained version of our approach fails. In both cases, the saliency images have been generated using our Approach B. A possible explanation for the failures in this latter case could be that the saliency images are not able to capture the relevant region of the image for fine-grained discrimination. Thus, the saliency-modulated layer focuses on the wrong features for the task. We also wanted to show our saliency map and compare it with 2 of the algorithms used in this thesis for the offline generation of saliency map, which are ISEEL and SALICON, these examples can be seen in the Table 4.8

4.4 Conclusion

In this chapter, we proposed a method to improve fine-grained image classification by means of saliency maps. Our method does not require explicit saliency maps, but they are learned implicitly during the training of an end-to-end deep convolutional network. We validated our method on several datasets for fine-grained classification tasks (Flowers, Birds and Cars). We showed that our approach obtains similar results as the SMIC method, which required explicit saliency maps. We showed

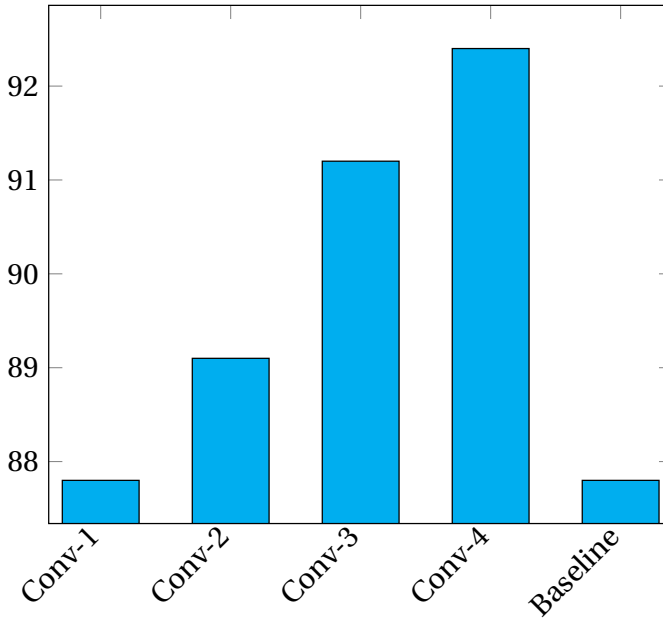


Figure 4.2: Graph shows the classification accuracy on Flowers for various number of layers in the saliency branch. Best results are obtained with four convolutional layers. Baseline refers to the method without saliency branch.

that combining RGB data with saliency maps represents a significant advantage for object recognition, especially for the case when training data is limited.

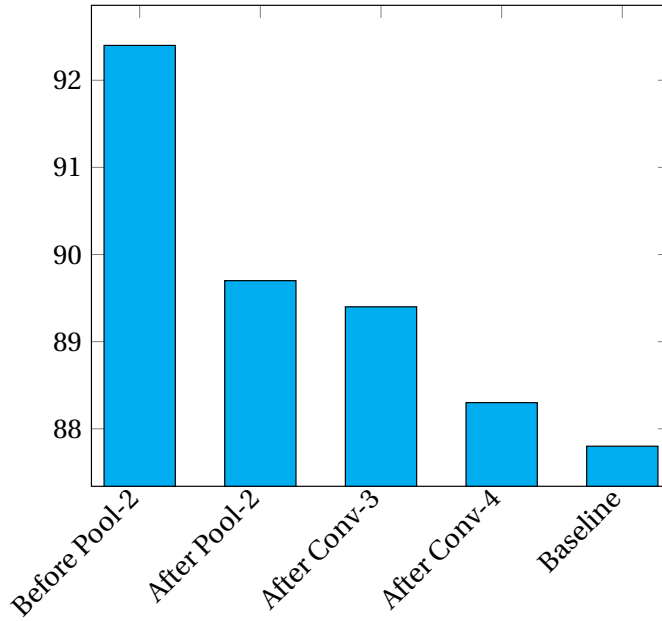


Figure 4.3: Graph shows the classification accuracy on Flowers. Various points for fusing the saliency and RGB branch are evaluated. Best results are obtained when fusion is placed before the pool-2 layer. Baseline refers to the method without saliency branch.

4.4. Conclusion

	#train images	1	2	3	5	10	15	20	25	30	K	AVG
Flowers	Baseline-RGB	31.8	45.8	53.1	63.6	72.4	76.9	81.2	85.1	87.2	87.8	68.3
	Baseline-RGB + scratch SAL	34.3	48.9	54.3	65.9	73.1	77.4	82.3	85.9	88.9	89.1	70.0
	SMIC [46]*	37.6	51.9	57.1	68.5	75.2	79.7	84.9	88.2	91.2	92.3	72.7
	Approach A	36.9	51.3	56.9	67.8	74.9	78.4	82.9	88.1	90.9	92.0	72.0
	Approach B	37.3	51.7	57.2	68.7	75.6	78.7	83.8	88.4	91.7	92.5	72.6
Cars	Baseline-RGB	4.1	7.8	11.7	17.3	25.5	31.1	38.5	42.2	47.2	60.0	28.5
	Baseline-RGB + scratch SAL	5.9	10.7	14.4	19.1	27.4	32.9	38.5	44.0	48.7	61.5	30.3
	SMIC [46]*	9.3	14.0	18.0	22.8	30.0	34.7	40.4	46.0	50.0	61.4	32.7
	Approach A	9.3	14.3	17.4	22.3	28.4	35.3	39.7	45.7	50.1	61.9	32.4
	Approach B	9.8	15.1	18.4	22.9	28.8	35.1	39.9	45.8	49.7	62.9	32.8
Birds	Baseline-RGB	9.1	13.6	19.4	27.7	37.8	44.3	48.0	50.0	54.2	57.0	34.8
	Baseline-RGB + scratch SAL	10.4	14.9	20.3	28.3	38.6	43.9	46.9	48.4	50.7	55.7	35.8
	SMIC [46]*	13.1	18.9	22.2	30.2	38.7	44.3	48.0	50.0	54.2	57.0	37.7
	Approach A	11.8	18.3	22.1	29.3	39.1	44.4	47.8	49.7	53.1	56.5	37.2
	Approach B	12.9	18.7	22.7	29.7	39.4	44.1	48.2	49.9	53.9	57.7	37.7

Table 4.3: Classification accuracy for Flowers, Cars, and Birds dataset (results are the average over three runs), using **AlexNet** as base network. Results are provided for varying number of training images, from 1 until 30; K refers to using the number of training images used in the official dataset split. The rightmost column shows the average. The * indicates that the method requires an explicit saliency method. Our method (Approach B) obtains similar results as SMIC but without the need of a pretrained saliency network trained on a saliency dataset.

	#train images	1	2	3	5	10	15	20	25	30	K	AVG
Flowers	Baseline-RGB	39.0	60.1	68.0	82.5	89.0	92.0	92.1	93.3	94.2	95.4	80.3
	Baseline-RGB + scratch SAL	40.1	63.8	69.7	83.9	89.7	91.9	92.9	93.8	95.1	97.1	81.8
	SMIC [46]*	42.6	64.2	70.9	85.5	90.9	92.7	94.0	95.0	97.0	97.8	83.1
	Approach A	42.4	64.5	70.7	85.2	90.3	92.4	93.3	94.3	96.5	97.9	82.8
	Approach B	42.7	64.5	71.0	85.1	90.4	92.5	93.1	94.7	96.8	98.1	82.9
Cars	Baseline-RGB	30.9	45.8	53.1	62.7	70.9	73.9	79.9	88.7	89.2	90.7	68.6
	Baseline-RGB + scratch SAL	33.8	46.1	54.8	63.8	71.7	74.9	80.9	88.1	89.1	91.0	69.4
	SMIC [46]*	34.7	47.9	55.2	64.9	72.1	75.8	82.1	90.0	91.1	92.4	70.6
	Approach A	34.1	47.0	56.3	64.7	71.9	75.3	81.7	89.0	90.8	91.7	70.2
	Approach B	34.0	47.5	55.4	64.7	71.8	75.5	81.9	89.3	91.0	92.1	70.3
Birds	Baseline-RGB	24.9	35.3	44.1	53.3	63.8	71.8	75.7	79.3	82.9	83.7	61.5
	Baseline-RGB + scratch SAL	26.3	36.1	45.2	53.9	64.3	72.1	76.3	79.9	83.1	83.4	62.1
	SMIC [46]*	28.1	37.9	46.8	55.2	65.3	73.1	77.0	82.9	84.4	86.1	63.7
	Approach A	26.9	36.9	46.1	54.2	64.9	72.8	77.1	81.4	83.4	84.8	62.9
	Approach B	27.1	37.0	46.2	54.9	65.4	72.8	77.1	81.3	83.8	85.1	63.1

Table 4.4: Classification accuracy for Flowers, Cars, and Birds dataset (results are the average over three runs), using **ResNet152** as base network. Results are provided for varying number of training images, from 1 until 30; K refers to using the number of training images used in the official dataset split. The rightmost column shows the average. The * indicates that the method requires an explicit saliency method. Our method (Approach B) obtains similar results as SMIC but without the need of a pretrained saliency network trained on a saliency dataset.

Chapter 4. Hallucinating Saliency Maps for Fine-Grained Image Classification for Limited Data Domains

Method	Flowers	Birds	Cars
Krause et al. [91]	-	82.0	92.6
Bilinear-CNN [109]	-	84.1	91.3
Compact Bilinear Pooling [51]	-	84.3	91.2
Low-rank Bilinear Pooling [88]	-	84.2	90.9
Cui et al. (with Imagenet) [32]	96.3	82.8	91.3
MA-CNN [227]	-	86.5	92.8
Ge-Yu [53]	90.3	-	-
DLA [210]	-	85.1	94.1
SMIC [46]*	97.8	86.1	92.4
Approach A	97.3	84.8	91.7
Approach B	97.9	85.1	92.1

Table 4.5: Comparison with state of the art methods for domain-specific fine-grained recognition using the standard data splits of Flowers, Birds and Cars.

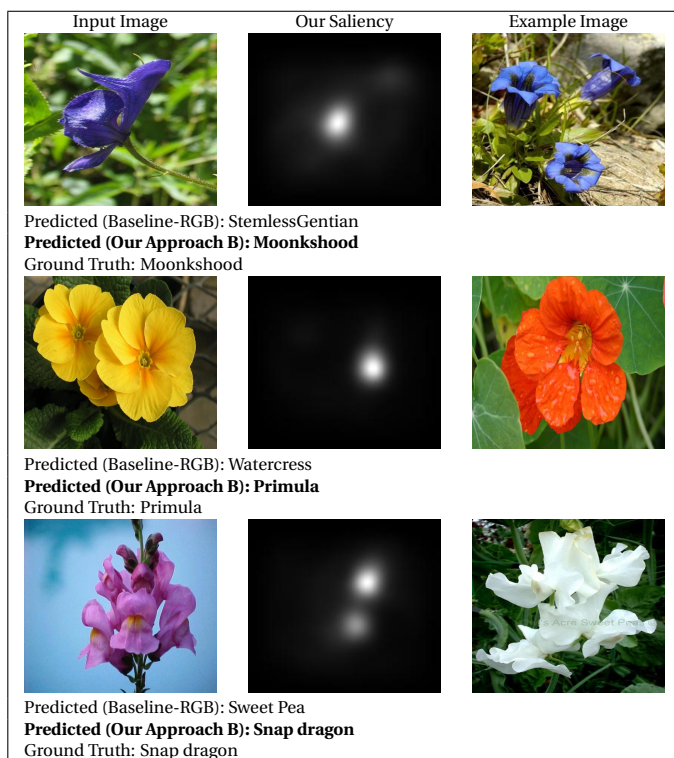


Table 4.6: Some success examples on Flowers: when the prediction done by Baseline-RGB fails to infer the right label for some test images, but the prediction by our approach is correct. Example image contains image of the wrongly predicted class.

Chapter 4. Hallucinating Saliency Maps for Fine-Grained Image Classification for Limited Data Domains

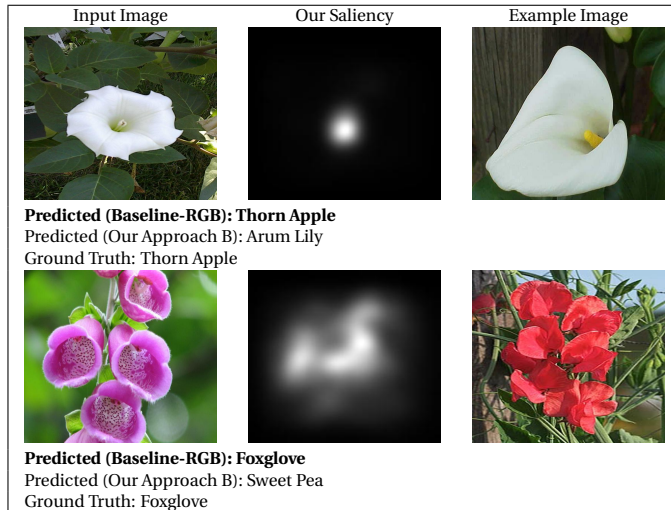


Table 4.7: Some failure examples on Flowers: when the prediction done by our method fails to infer the right label for some test images, but the prediction by Baseline-RGB is correct. Example image contains image of the wrongly predicted class.


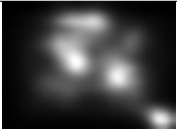

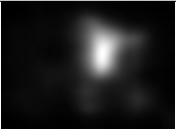





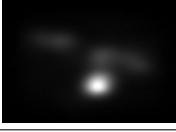
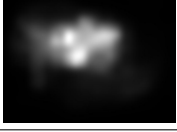

Dataset	RGB Original	Our Approach B	ISEEL	SALICON
Flowers-102				
				
				

Table 4.8: Visualization of saliency maps for Flowers-102, Birds and Cars.

5 Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition¹

5.1 Introduction

One of the perceptual cues used for scene understanding is image saliency, i.e. a representation of the scene that highlights those regions which are more informative than their surroundings. Computational methods in saliency detection used in computer vision are intended to determine which regions of the image attract humans' attention. Saliency methods can be divided in two main categories: (i) salient object detection methods (which segment relevant objects in the image) [222, 224]; and (ii) methods which produce eye-fixation maps [72, 129, 138]. For the second category, which is the focus of this chapter, the common way to obtain an accurate saliency map is to perform eye tracking experiments on still images. Eye fixations from different participants are fused to obtain a unique map, named fixation map, which will represent the saliency ground truth.

In [76] proposed one of the first computational saliency methods based on combining the saliency cues for color, orientation and luminance. Many works followed proposing a large variety of hand-crafted features for saliency [11, 146]. In the last decade, computational saliency estimation has moved from handcrafted to deep features [106]. These methods aim to find a network that computes saliency maps that are close to ground truth saliency maps. A limitation of these approaches is that they require saliency ground truth for their training. Generating saliency

¹This chapter has been submitted to a journal.

Chapter 5. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition

ground truth is a costly process and is required for each new dataset, and affects the efficiency of these approaches.

In the human visual system, saliency is applied to select a small part of the incoming sensory information. As a result massive sensory input can be processed despite limited computational capacity of the brain [75]. It allows humans to rapidly and efficiently process the incoming information. The capability to attend the most relevant information in the image present in the human visual system could also be important for neural networks that aim to process visual data. In this chapter, we endow a neural network that aims to perform object recognition with a separate branch that computes a saliency map. This map is used to attend to specific regions in the image (thereby selecting the part of the information deemed most relevant). The potential of such a network is that it can be trained on any image classification dataset. The saliency maps would be the side-effect of training this network, and hence our method allows for the computation of saliency without needing any eyetracking ground truth data to train.

The ground truth of saliency is obtained by locating fixations in the scene. These (binary) fixation maps are then smoothed by 1 degree of visual angle (dva or deg) in order to simulate the average deviation of capture of the eye tracker [101][177]. This smoothing is usually done using a circular gaussian filter, obtaining a continuous representation of the saliency map. Several image processing and computer vision techniques have been used in order to accurately represent saliency maps.

The saliency map is assumed to be specific for each image (depending on image features), but experimentation may induce certain patterns such as the center bias. The center bias (CB) is the common region where participants tend to look, this can be due to: (i) photographs tend to frame the salient object centered on the image, (ii) there are oculomotor tendencies from the task focusing the gaze on the center [131] and (iii) some images do not show objects salient enough to focus attention outside the center. This center bias is present in most saliency datasets and is also exploited by several saliency models to better simulate human data.

In this chapter we evaluate the accuracy of the saliency maps that are produced as a side-effect of object recognition. Additionally, we also evaluate the usage of supervised and unsupervised center bias (CB) in our framework. We show that the CB improves in most datasets where the CB is more present. To summarize, our main contributions are:

- We demonstrate that it is possible to obtain accurate saliency maps by training an object recognition network endowed with a saliency branch. Our method

does not require any saliency ground truth data.

- We include an extensive study of the effect of center bias on the results.
- Extensive experiments performed on real and synthetic image datasets show that highly accurate saliency maps are obtained. Our method obtains competitive results on several standard benchmark datasets and the new state-of-the-art on the CAT2000 dataset.

The current chapter is related to previous chapter. There we focus on fine-grained image classification, and show that a saliency branch can be used to improve results. In this chapter, we show that a saliency branch trained for image classification can actually obtain competitive results on the saliency benchmark dataset, without requiring any saliency ground truth data for training. To the best of our knowledge, we are the first to show that saliency prediction can be obtained as a side-effect of object recognition.

5.2 Proposed Approach

In the current chapter, we extend the work in the previous chapter by including a study of the center bias which allows us to further improve the results. In addition, we perform an analysis of the quality of the saliency estimation on three new datasets (CAT2000, MIT1003 and KTH).

5.2.1 Network architecture

The overview of our proposed method is shown in Figure 5.1. The network consists of two branches: one to extract the features from an RGB image (the red branch called *RGB branch*), and the other one (called the *saliency branch* marked in green) to generate the saliency maps from the same RGB image. Both branches are combined using a *modulation layer* (represented by the \times symbol) and the output is further processed by several shared layers ending up with a classification layer.

Consider an input image $I(x_1, x_2, x_3)$, where x_1, x_2 are the spatial coordinates and $x_3 = \{1, 2, 3\}$ indicate the three color channels of the image. Let us define the three networks as being s for the saliency branch, r for the RGB branch and f for the final shared layers. We will name the output of the saliency branch the saliency image $S(x_1, x_2)$ (we will design the saliency branch to output only a single saliency image, therefore there are only two coordinates involved), and the output of the

Chapter 5. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition

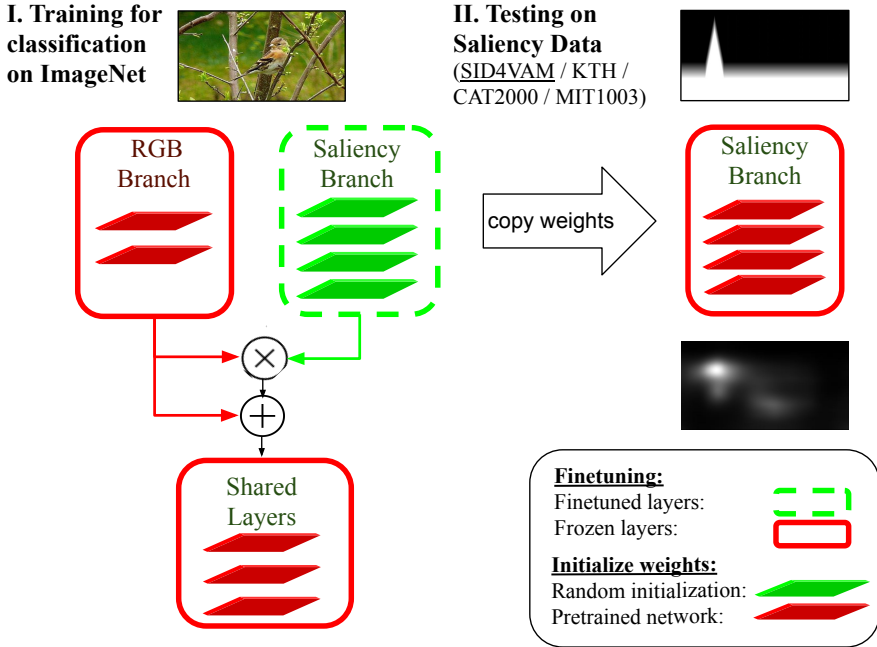


Figure 5.1: Overview of our method. We process an RGB input image through two branches: one branch extracts the RGB features and the other one is used to learn saliency maps.

RGB branch $R(x_1, x_2, x_3)$. Both S and R will have the same spatial resolution. We now define the modulation layer as:

$$\begin{aligned} \hat{R}(x_1, x_2, x_3) &= r(I(x_1, x_2, x_3)) \cdot (s(I(x_1, x_2, x_3)) + 1) \\ &= R(x_1, x_2, x_3) \cdot S(x_1, x_2) + R(x_1, x_2, x_3). \end{aligned} \quad (5.1)$$

Note that the same saliency branch output S is applied to all the feature maps of R (along the x_3 dimension). The output \hat{R} is a summation of the modulated output $R \cdot S$ and a non-modulated version of the RGB branch R (see also the skip connection represented by \oplus in Figure 5.1). This was found to improve results in [46]. The output of the modulation layer is then used as an input to the shared

layers to obtain the final prediction over the classes y :

$$p(y|I) = f(\hat{R}). \quad (5.2)$$

where we omit the spatial coordinates for clarity. We train the network for the task of image classification on a training dataset \mathcal{D} of images with the cross-entropy loss:

$$\mathcal{L} = \sum_{I \in \mathcal{D}} \log p_{c(I)}(y|I) \quad (5.3)$$

where \mathcal{D} is the entire training dataset and $c(I)$ is the ground truth label of image I and p_c is the c -th element of the vector p .

The RGB branch followed by the modulation layers resembles a standard image classification network (see layers marked in red in the Figure 5.1-left). In this work, we will consider several architectures, including AlexNet [93], VGG16 [162], and ResNet152 [62]. The saliency branch consists of four convolutional layers, similar to the first three layers of the AlexNet architecture combined with a 1×1 convolutional layer. More precisely, the output of the third convolutional layer, i.e. the one with 384 dimensional feature maps with a spatial resolution of 13×13 (for a 227×227 RGB input image), is further processed using a 1×1 convolution and then a ReLU activation function. This 1×1 convolution maps the feature map to a single output feature map, and its goal is to calculate the score for each "pixel" and to produce a single map that can be used to modulate the RGB branch. Finally, to generate the input for the posterior classification network, the 13×13 saliency maps are upsampled at 27×27 (which is the default input size of the following classification module) through bilinear interpolation.

What differentiates our architecture from a standard object recognition network, is the introduction of the saliency branch which transforms the RGB input image into a *modulation map* S . While training the network the modulation map learns to focus on those features that are important to perform the classification task. This is a very similar task as for which the human visual system is thought to use visual saliency, namely to identify those regions of high information in the image. In this chapter, we show that this modulation map resembles a saliency map. Actually when compared to saliency maps obtained from human eye-tracking experiments, this modulation map is found to provide a surprisingly good estimate of them.

5.2.2 Training the saliency branch

Our approach is depicted in Figure 5.1. The main idea is to train the saliency branch S on a classification task. By optimizing the network to be good in image classification, we hypothesize that the saliency branch will learn a mapping from the image I to something similar as a saliency map. The modulation map S will provide higher values to those regions that are important to performing the image classification task. The learned network s will then be evaluated on several existing saliency estimation datasets. Interestingly, the network s has not been trained with any saliency ground truth, rather the saliency network is trained as a side-effect of training a network optimal for object recognition.

We would like the classification task to be very general to ensure that the saliency network is trained on a wide variety of images. We therefore choose to train the network on the ImageNet dataset [93] which has 1000 different classes, including classes from plants, sports, artefacts, animals, etc.

As explained above, the purpose of the saliency branch is to generate a saliency map directly from an RGB input image. This network is built by initializing the RGB branch with ImageNet pre-trained weights. The weights of the saliency branch are initialized randomly using the Xavier method [56] (see Figure 5.1, green layers). Then, the network is selectively trained: we allow to train only the layers corresponding to the Saliency branch (represented by the surrounding green dotted line box) and to freeze all the remaining layers (represented by the solid red line boxes). During training, the saliency branch learns to focus on those regions of the image that are important for the classification of the 1000 ImageNet classes.

Once the Imagenet training is finished, we only use the saliency branch, freeze its weights, and test it on the images of various saliency estimation datasets (see Figure 5.1-right). We will consider both datasets with real images (Toronto [18], MIT1003 [81], KTH [89]) as well as datasets that contain synthetic images (CAT2000 [13] and SID4VAM [9]).

5.2.3 Combination with center bias

Eye movement datasets used for saliency evaluation tend to be center biased (namely, that most fixations tend to be at the center of the image, see Table 5.11). Several factors on the experimentation and the stimuli can cause this effect. For instance, most real images frame the scene (the relevant or salient part is in the center of view in photographs). Non-salient/non-popout stimuli [9, 171, 185] has

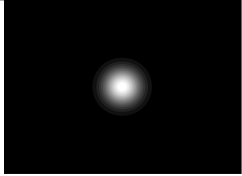
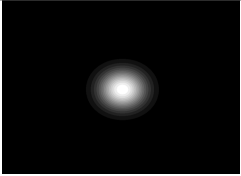
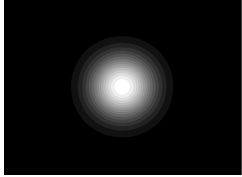
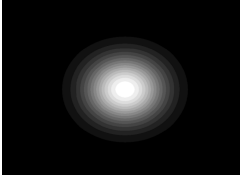
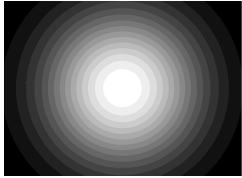
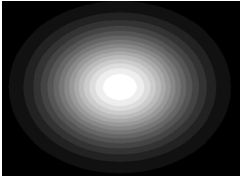
GVA	Circular	Ellipsoid
36 x 2		
36 x 5		
36 x 14		

Table 5.1: Simulating the Center Bias by parametrizing Gaussian

been shown to promote center biases, as participants do not have any region to attend to, specially if the task sometimes involves centering the gaze on the image. These center biases have an influence on how to evaluate saliency models upon predicting fixations [14, 21], as these fixations are accounted while are not specific to image saliency.

To compute the center bias, we use an isotropic 2d gaussian low-pass filter with $\sigma = \text{GVA} / (2\sqrt{2\log(2)})$, with a window of $6\sigma \times 6\sigma$. Using a parameter "GVA" as a multiplying factor of the pixels per degree of visual angle. This is the usual smoothing function [19, 20, 21] for building the fixation density maps.

Supervised CB (SCB) With the gaussian function can compute the exact center bias from data with 1 deg of GVA and overlapping all binary fixation maps (seemingly obtaining a unique fixation density map with images put altogether). For evaluation, we split the data in two sets, generating the center bias for each of them and evaluating each sample with the opposite split. See in Table 5.11 (column 2)

examples of center biases for different datasets.

Unsupervised CB (UCB) For the CB we used circular (UCBc) and ellipsoid (UCBe) versions of the gaussian function (see Table 5.1). We did this as center biases might vary on the display and experimental methods for each saliency dataset, in most cases these maps are spread in the horizontal axis, as image is usually of rectangular shape. For the ellipsoid case, we resized the image so that the resulting map is stretched horizontally with a factor of +50%, but keeping the same GVA vertically.

We selected for the GVA according to the following rule: 2 deg corresponds to the approximate maximum diameter of coordinate deviation permitted during eye tracking calibration, this is approximately 2 times the common deviation of participant's fixations [101, pp 255]; [177, pp 778], 5 deg corresponds to the degrees of higher visual acuity of foveal/central vision [166] and 14 deg corresponds to the radius of the screen (about 512px).

Fusion Previously, other models (see Table 5.2 - column 7) used additional computations from priors or baselines from fixation data. For instance, DeepGazeII summed the center baseline whereas ML-Net and SAM the learned priors are used for modulating the result of the network. We defined two regimes for fusing the RGB and the saliency branch: sum (CB^+) or multiplication (CB^\times). With this we can test at distinct baselines the effect of the center bias over the saliency map produced by the network. See in Table 5.10 different examples of the resulting fusion (sum or multiplication).

5.3 Experimental Results

5.3.1 Setup

Datasets

We have computed the saliency maps for images from distinct eye-tracking dataset, corresponding to 120 real scenes (Toronto) [18], 40 nature scenes (KTH) [89], 100 synthetic patterns (CAT2000) [13] and 230 synthetic images with specific feature contrast (SID4VAM) [9]. We have computed these images images dataset with our approach, supervised artificial model that specifically compute high-level features (DeepGazeII, ML-Net, SAM, salGAN), and models biological inspiration

5.3. Experimental Results

#	Name	Year	Features/Architecture	Mechanism
1	IKN	1998	DoG (color+intensity)	-
2	AIM	2005	ICA (infomax)	max-like
3	GBVS	2006	Markov chains	graph prob.
4	SDLF	2006	Steerable pyramid	local+global prob.
5	ML-Net	2016	VGG-16	Backprop.(finetuning)
6	DeepGazeII	2016	VGG-19	Backprop.(finetuning)
7	SAM	2018	VGG-16/ResNet-50+LSTM	Backprop.(finetuning)
8	SalGAN	2017	VGG-16 Autoencoder	Finetuning+GAN Loss
#	Name	Learning	Training Data (#img)	Bias/Priors
1	IKN	-	-	-
2	AIM	Unsupervised	Corel (3600)	-
3	GBVS	Unsupervised	Einhauser (108)	graph norm.
4	SDLF	Unsupervised	Oliva (8100)	scene priors
5	ML-Net	SALICON (10k), MIT (1003)	learned priors	-
6	DeepGazeII	Supervised	SALICON (10k), MIT (1003)	center bias
7	SAM	Supervised	SALICON (10k) & others	gaussian priors
8	SalGAN	Supervised	SALICON (10k), MIT (1003)	-

DoG: difference of gaussians, ICA: independent component analysis, C-S: center-surround, max-like: max-likelihood probability, BCE: binary cross-entropy, GAN: Generative adversarial network

Table 5.2: Description of saliency models

Dataset	Type	# Images	# PP	pxva	Resolution
TORONTO	Indoors & Outdoors	120	20	32	681x511
MIT1003	Indoors & Outdoors	1003	15	35	1024x768
KTH _n	Nature photos	99	31	34	1024x768
CAT2000 _p	Synthetic Patterns	100	18	38	1920x1080
SID4VAM	Synthetic Pop-out	230	34	40	1280x1024

pxva: pixels per 1 degree of visual angle, PP: participants

Table 5.3: Characteristics of eye tracking datasets

(IKN, AIM, SDLF and GBVS).

Base-Networks

We evaluate our approach using three network architectures: AlexNet [93], which consists of five convolutional layers followed by three fully connected layers, VGG16 and ResNet-152 [62], consisting of 152 convolutional layers, respectively, organized in 5 residual blocks

5.3.2 Results

First experiment: multiple networks.

In order to evaluate how accurate the saliency map is able to match the location of human fixations, we used a set of metrics previously defined by [15] and [21]

In Table 5.4 we show results of Area Under ROC (AUC), Correlation Coefficient (CC), Normalized Scanpath Saliency (NSS), Kullback-Leibler divergence (KL), similarity (SIM) for every network for all datasets.

The area under ROC (AUC) considers as true positives the saliency map values that coincide with a fixation and false positives the saliency map that have no fixation, then computes the area under the curve. Similarly, the NSS computes the average normalized saliency map that coincide with fixations. Other metrics such as CC, KL and SIM compute the score upon the region distribution statistics of all pixels (KL calculating the divergence and CC/SIM the histogram intersection or similarity of the distribution).

Dataset	Model	AUC-Judd	AUC-Borji	CC	NSS	KL↓	SIM
MIT1003	AlexNet	0.7323	0.7034	<u>0.2597</u>	<u>0.8654</u>	<u>1.7622</u>	0.2844
	VGG16	<u>0.7402</u>	<u>0.7199</u>	0.2594	0.8597	1.7772	<u>0.2899</u>
	ResNet152	0.7231	0.7084	0.2531	0.8550	2.0785	0.2839
TORONTO	AlexNet	0.7679	0.7308	0.4546	1.3718	<u>1.5134</u>	0.3944
	VGG16	0.7812	<u>0.7475</u>	0.4627	1.4045	1.5179	0.4201
	ResNet152	<u>0.7816</u>	0.7323	<u>0.5378</u>	<u>1.6433</u>	1.6991	<u>0.4390</u>
KTH	AlexNet	0.5975	0.5881	0.2249	0.3374	<u>1.0083</u>	<u>0.5112</u>
	VGG16	0.6028	0.5793	0.2250	0.3459	1.3194	0.4848
	ResNet152	<u>0.6154</u>	<u>0.5869</u>	<u>0.2942</u>	<u>0.4436</u>	1.3492	0.4989
CAT2000	AlexNet	0.7005	0.6710	0.2950	0.7468	1.4615	0.3936
	VGG16	0.7113	0.6741	<u>0.3151</u>	0.8371	1.4510	0.4031
	ResNet152	<u>0.7217</u>	<u>0.6805</u>	0.3100	<u>0.8548</u>	<u>1.2876</u>	<u>0.4064</u>
SID4VAM	AlexNet	<u>0.7413</u>	<u>0.7216</u>	<u>0.3889</u>	<u>1.4256</u>	<u>1.6652</u>	<u>0.4085</u>
	VGG16	0.6752	0.6506	0.2707	0.8477	1.9129	0.3695
	ResNet152	0.6988	0.6723	0.3010	1.1140	1.9790	0.3786

Table 5.4: Benchmark of our method with different networks (top-1 networks are underlined)

After computing the saliency maps for all datasets (see in 5.4) with AlexNet, VGG16 and ResNet152 we observed that metric scores vary considerably depending

on dataset or network. AlexNet is shown to provide best results for pop-out patterns (SID4VAM) whereas ResNet152 and VGG16 showed overall higher scores with real images (MIT1003, TORONTO, KTH).

Second experiment: Center bias analysis

In this section we wanted to analyze the use of the center bias, for this we have carried out different experiments. To begin with, we wanted to test different ways of normalization (Min-Max, Energy and Standarization) of our center bias, we can see this in the Table 5.5.

Adding, we perform a random permutation of images for train and test (i.e. 50% and 50%). For example, in the case of Toronto, there are 120 images, as we should randomly select 60 images for train (T1) and 60 images for test (T2). So it would be interesting to combine the density map with the baseline of each different split, that is, first we generate the center bias baseline of T1 and T2, then you take your saliency maps (from split T1) you combine them (mult / sum) with the T2 baseline and then vice-versa: using the T2 saliency maps combined with the T1 baseline. The results of this experiment are observed in Table 5.6

In addition, we have used the original pxva of the five datasets (See Table 5.7 and Table 5.8) and we have compared the different sizes of the single Gaussian in its two forms (circular and elliptical), for which we have used only the Min-Max normalization for the fusion type Sum or Multiplication.

We have also performed an ablation study for evaluating the effect of the center bias and the fusion. From this, we tested the center bias extracted from the data (SCB) as well as our unsupervised implementation using gaussian and ellipsoid baselines (UCBc and UCBe), testing both fusions of Mult-MinMax or Sum-MinMax. For most datasets, the UCBe gave highest scores using a GVA factor of 14 deg and the Sum-MinMax fusion (adding the baseline to the saliency map). For the cases of SCB, the Mult-MinMax scored higher, although both fusions gave very similar scores.

In Table 5.10 we show results for our model with different parametization and using a distribution-based metrics; AUC-Judd and SIM.

Third experiment: State of the Art.

In Table 5.15 and Table 5.14 we show results for our model with different parametization and our best setting in comparison with the state of the art saliency models

Chapter 5. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition

(using both location- and distribution-based metrics; AUC-Judd and SIM).

Fourth experiment: Visual results and center bias

We can see in Table 5.11 different examples of images and the generated saliency maps from different scenes (one per each dataset), including an illustration of each dataset center bias. We can observe that the fusion is able to modulate the saliency map, showing that it can be better to use one strategy or another.

Fifth experiment: Qualitative results

These saliency prediction results show that our model has robust metric scores on both real images and synthetic images for saliency prediction. Again, we would like to stress that our model is not trained on fixation prediction datasets and does not add a center Gaussian to leverage some metrics due to the center bias. Our model performs best on detecting pop-out effects (from visual attention theories [76]), whilst performing similarly for real image datasets (Figure 5.2). Some deep saliency models use several mechanisms to leverage (or/and train) performance for improving saliency metric scores, such as smoothing/thresholding (see Figure 5.2, rows 4-5) or a center gaussian (see Figure 5.3, row 5). We also consider that some of these models are already finetuned for synthetic images (e.g. SAM-ResNet [30]). *Our Approach* (that has not been trained in these type of datasets) has shown to be robust on these two distinct scenarios/domains.

Sixth experiment: Evaluation benchmark of saliency hallucination

Here we compare the saliency estimation which is obtained after only performing Step I in Figure 5.1 with existing saliency methods. This saliency estimation is trained without access to any ground truth saliency data, and is obtained while training the image classification task on Imagenet.

Saliency prediction metrics assign a score depending on how well the predicted saliency map is able to match with locations of human fixations (see definitions in Borji et al.[15] and Bylinskii et al.[21]). We selected the Area Under ROC (AUC), Kullback-Leibler divergence (KL), similarity (SIM), shuffled AUC (sAUC) and Information Gain (IG) metrics considering its consistency of predictions of human fixation maps as well as towards to the center bias. We compare scores with classical saliency models, both with handcrafted low-level features (i.e. IKF [76], AIM [18], SDLF [177] and GBVS [60]) and state-of-the-art deep saliency models (i.e. DeepGazeII [97], SAM-ResNet [30], SALICON [72, 175] and SalGAN [137]) mainly pretrained on human fixations. The results are surprising, our method which has


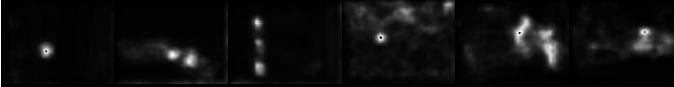
Model						
Humans						
GBVS						
OpenSALICON						
SAM-ResNet						
<i>Our Approach</i>						

Figure 5.2: Qualitative results for real images (Toronto dataset). Each image is represented in a different column and each model saliency map in each row. The ground truth density map of human fixations is represented in the 2nd row.

not been trained on any saliency data obtains competitive results. For the case of *Toronto* (Table 5.12) the best models are GBVS and OpenSALICON, followed by our model that scores in the top-3 of KL and SAM-ResNet that scores slightly higher in InfoGain metric. For the case of *SID4VAM* (Table 5.13) our approach gets best scores for most metrics compared to other deep saliency models, being mainly the top-2 acquiring similar scores to GBVS in most metrics (outperforming it in AUC measures).

Finally, we have compared scores with classical unsupervised saliency models (i.e. IKN, AIM, SDLF and GBVS) and state of the art deep saliency models (i.e. ML-Net, DeepGazeII, SAM and SalGAN), mainly backpropagating scores on human fixation data. Our model outperforms other unsupervised saliency models in both AUC and SIM metrics (see Table 5.15 and Table 5.14, respectively), and outperforms all other deep saliency models with synthetic and pop-out patterns (CAT2000, SID4VAM). This suggests that we are able to extract bottom-up attention but we are not biased to specific features of the dataset. Accounting that our model is not trained on human fixation data, our model scores top-3 in AUC and SIM metrics for real image saliency datasets (TORONTO, MIT1003, KTH).

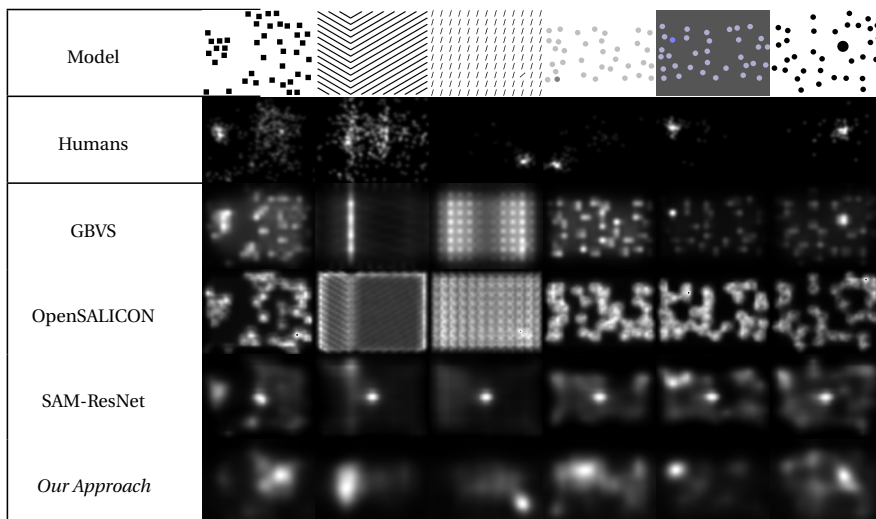


Figure 5.3: Qualitative results for synthetic images (SID4VAM dataset). Each image is represented in a different column and each model saliency map in each row. The ground truth density map of human fixations is represented in the 2nd row.

5.4 Conclusion

This chapter shows that saliency might be an intrinsic effect in image representation learning, and this can be obtained by training other tasks such as image classification. By training on ImageNet classification we are able to extract saliency maps and modulate that factor to a specific center bias. Our model appears to be robust with different metrics and datasets, outperforming classical unsupervised models and with a trend to acquire similar results to the state of the art, even with limited data. We have added a study of which networks and typologies of center biases can affect saliency prediction. Our results showed that scores vary considerably depending on dataset or network, as every dataset has specific set of features and parametrization of experimental systematic tendencies, suggesting that there cannot be a unique solution for modeling the center bias in combination with saliency. Possible improvements could include finetuning with fixation data, abling to tune the saliency branch (and/or the center bias) by training on fixation data.

5.4. Conclusion

Dataset	Fusion	Normalization	AUC-Judd	AUC-Borji	CC	NSS	KL↓	SIM
MIT1003	Mult	MinMax	0.798	0.738	0.358	1.182	1.797	0.357
		Energy	0.796	0.737	0.357	1.182	1.796	0.357
		Standarization	0.797	0.742	0.356	1.179	1.788	0.355
	Sum	MinMax	0.795	0.786	0.364	1.186	1.517	0.311
		Energy	0.793	0.785	0.364	1.185	1.517	0.310
		Standarization	0.794	0.785	0.364	1.185	1.517	0.310
Toronto	Mult	MinMax	0.796	0.714	0.480	1.445	2.499	0.439
		Energy	0.795	0.713	0.478	1.444	2.498	0.438
		Standarization	0.794	0.714	0.479	1.444	2.498	0.438
	Sum	MinMax	0.793	0.778	0.464	1.355	1.220	0.393
		Energy	0.792	0.776	0.463	1.356	1.221	0.392
		Standarization	0.792	0.777	0.464	1.356	1.221	0.390
KTH	Mult	MinMax	0.628	0.596	0.274	0.396	1.300	0.511
		Energy	0.626	0.597	0.273	0.394	1.299	0.511
		Standarization	0.627	0.596	0.273	0.393	1.300	0.510
	Sum	MinMax	0.634	0.629	0.326	0.462	0.667	0.560
		Energy	0.631	0.627	0.322	0.461	0.669	0.559
		Standarization	0.633	0.627	0.323	0.460	0.668	0.558
CAT2000	Mult	MinMax	0.812	0.735	0.655	1.688	1.540	0.560
		Energy	0.811	0.734	0.654	1.687	1.539	0.561
		Standarization	0.811	0.733	0.654	1.686	1.539	0.560
	Sum	MinMax	0.787	0.775	0.518	1.321	0.992	0.451
		Energy	0.785	0.773	0.518	1.322	0.991	0.452
		Standarization	0.786	0.774	0.518	1.320	0.990	0.451
SID4VAM	Mult	MinMax	0.746	0.708	0.383	1.375	2.142	0.403
		Energy	0.745	0.707	0.382	1.373	2.143	0.401
		Standarization	0.745	0.707	0.382	1.373	2.143	0.402
	Sum	MinMax	0.742	0.735	0.338	1.043	1.474	0.381
		Energy	0.741	0.733	0.337	1.041	1.475	0.380
		Standarization	0.741	0.734	0.338	0.043	1.474	0.381

Table 5.5: Analysis of **normalization** on MIT1003, Toronto, KTH, CAT2000 and SID4VAM, using AlexNet as base-network

Chapter 5. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition

Dataset	Fusion	Normalization	AUC-Judd	AUC-Borji	CC	NSS	KL↓	SIM
MIT1003	Mult	MinMax	0.796	0.738	0.358	1.099	1.737	0.355
		Energy	0.795	0.737	0.357	1.182	1.797	0.357
		Standarization	0.794	0.737	0.336	1.180	1.797	0.356
	Sum	MinMax	0.795	0.786	0.364	1.186	1.515	0.311
		Energy	0.794	0.785	0.364	1.185	1.516	0.310
		Standarization	0.792	0.785	0.363	1.185	1.516	0.310
Toronto	Mult	MinMax	0.796	0.715	0.479	1.445	2.499	0.439
		Energy	0.795	0.713	0.479	1.444	2.498	0.439
		Standarization	0.795	0.713	0.477	1.444	2.499	0.438
	Sum	MinMax	0.792	0.777	0.465	1.356	1.221	0.393
		Energy	0.791	0.776	0.464	1.356	1.220	0.393
		Standarization	0.791	0.776	0.464	1.355	1.221	0.393
KTH	Mult	MinMax	0.629	0.597	0.274	0.397	1.300	0.511
		Energy	0.628	0.596	0.274	0.395	1.299	0.512
		Standarization	0.627	0.596	0.274	0.395	1.301	0.511
	Sum	MinMax	0.634	0.629	0.323	0.466	0.669	0.559
		Energy	0.633	0.627	0.322	0.464	0.66	0.555
		Standarization	0.632	0.628	0.322	0.464	0.668	0.554
CAT2000	Mult	MinMax	0.813	0.735	0.655	1.688	1.541	0.560
		Energy	0.811	0.733	0.651	1.688	1.540	0.560
		Standarization	0.811	0.733	0.653	1.687	1.541	0.559
	Sum	MinMax	0.789	0.775	0.518	1.321	0.992	0.452
		Energy	0.787	0.777	0.518	1.322	0.993	0.451
		Standarization	0.788	0.776	0.518	1.322	0.992	0.452
SID4VAM	Mult	MinMax	0.747	0.708	0.383	1.377	2.142	0.404
		Energy	0.744	0.707	0.383	1.375	2.142	0.404
		Standarization	0.746	0.707	0.382	1.375	2.143	0.403
	Sum	MinMax	0.744	0.733	0.338	1.043	1.433	0.382
		Energy	0.742	0.733	0.336	1.043	1.475	0.381
		Standarization	0.741	0.734	0.337	1.043	1.474	0.381

Table 5.6: Analysis of **adquisition** on MIT1003, Toronto, KTH, CAT2000 and SID4VAM, using AlexNet as base-network

5.4. Conclusion

	Gaussian	GVA	Fusion	AUC-Judd	AUC-Borji	CC	NSS	KLJ	SIM
MIT1003	Circular	35 x 2	Multi-MinMax	0.595	0.548	0.196	0.667	1.807	0.154
			Sum-MinMax	0.7697	0.7264	0.4388	1.3115	1.5287	0.3987
		35 x 5	Multi-MinMax	0.718	0.632	0.304	1.014	0.906	0.302
			Sum-MinMax	0.768	0.740	0.343	1.133	1.616	0.322
		35 x 14	Multi-MinMax	0.792	0.743	0.349	1.152	1.679	<u>0.345</u>
			Sum-MinMax	<u>0.794</u>	0.785	<u>0.358</u>	<u>1.159</u>	1.547	0.301
	Ellipsoid	35 x 2	Multi-MinMax	0.651	0.581	0.252	0.858	1.372	0.230
			Sum-MinMax	0.758	0.722	0.320	1.061	1.666	0.309
		35 x 5	Multi-MinMax	0.724	0.639	0.315	1.013	<u>1.001</u>	0.3045
			Sum-MinMax	0.771	<u>0.751</u>	0.340	1.138	1.664	0.3200
		35 x 14	Multi-MinMax	0.799	0.750	0.351	1.168	1.700	0.350
			Sum-MinMax	0.800	0.750	0.360	1.169	1.699	0.338
Toronto	Circular	32 x 2	Multi-MinMax	0.616	0.564	0.232	0.698	1.752	0.193
			Sum-MinMax	0.770	0.726	0.439	1.312	1.529	0.399
		32 x 5	Multi-MinMax	0.762	0.673	0.431	1.309	6.637	0.391
			Sum-MinMax	0.781	0.753	0.447	1.311	1.492	0.406
		32 x 14	Multi-MinMax	<u>0.792</u>	0.740	<u>0.492</u>	<u>1.477</u>	1.689	<u>0.433</u>
			Sum-MinMax	0.790	<u>0.780</u>	0.430	1.238	1.344	0.338
	Ellipsoid	32 x 2	Multi-MinMax	0.640	0.657	0.354	1.018	1.500	0.314
			Sum-MinMax	0.776	0.740	0.430	1.238	1.344	0.338
		32 x 5	Multi-MinMax	0.780	0.681	0.438	1.311	1.511	0.401
			Sum-MinMax	0.789	0.757	0.450	1.374	1.542	0.402
		32 x 14	Multi-MinMax	0.800	0.741	0.497	1.500	1.701	0.440
			Sum-MinMax	0.801	0.780	0.439	1.301	1.350	0.340
KTH	Circular	36 x 2	Multi-MinMax	0.521	0.509	0.092	0.137	2.130	0.067
			Sum-MinMax	0.560	0.585	0.231	0.345	1.019	0.510
		36 x 5	Multi-MinMax	0.578	0.540	0.187	0.270	1.451	0.229
			Sum-MinMax	0.609	0.597	0.259	0.380	1.035	0.509
		36 x 14	Multi-MinMax	0.632	0.602	0.283	0.404	1.546	0.490
			Sum-MinMax	<u>0.635</u>	<u>0.630</u>	<u>0.327</u>	<u>0.465</u>	0.700	<u>0.556</u>
	Ellipsoid	36 x 2	Multi-MinMax	0.527	0.527	0.155	0.224	2.013	0.115
			Sum-MinMax	0.597	0.598	0.155	0.224	1.939	0.120
		36 x 5	Multi-MinMax	0.581	0.542	0.185	0.273	1.478	0.229
			Sum-MinMax	0.620	0.598	0.258	0.380	1.100	0.510
		36 x 14	Multi-MinMax	0.634	0.639	0.327	0.465	0.702	0.560
			Sum-MinMax	0.640	0.639	0.3266	0.466	<u>0.702</u>	0.561

Table 5.7: Analysis of **GVA gaussian on real images dataset**: MIT1003, Toronto and KTH.

Chapter 5. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition

	Gaussian	GVA	Fusion	AUC-Judd	AUC-Borji	CC	NSS	KL↓	SIM
CAT2000	Circular	38 x 2	Multi-MinMax	0.620	0.568	0.465	1.266	1.648	0.211
			Sum-MinMax	0.732	0.700	0.400	1.042	1.341	0.415
			Multi-MinMax	0.753	0.678	0.628	1.626	0.769	0.462
		38 x 5	Sum-MinMax	0.781	0.761	0.548	1.400	1.140	0.465
			Multi-MinMax	0.811	0.768	0.604	1.534	1.946	<u>0.545</u>
			Sum-MinMax	<u>0.819</u>	0.801	0.558	1.402	0.970	0.480
	Ellipsoid	38 x 2	Multi-MinMax	0.678	0.611	0.559	1.483	1.291	0.316
			Sum-MinMax	0.751	0.724	0.466	1.207	1.266	0.433
			Multi-MinMax	0.759	0.681	<u>0.627</u>	<u>1.600</u>	<u>0.887</u>	0.471
		38 x 5	Sum-MinMax	0.790	0.770	0.551	1.398	1.120	0.487
			Multi-MinMax	0.812	0.770	0.607	1.574	1.348	0.549
			Sum-MinMax	0.820	0.800	0.610	1.493	1.350	0.531
SID4VAM	Circular	40 x 2	Multi-MinMax	0.525	0.511	0.068	0.195	2.195	0.057
			Sum-MinMax	0.741	0.712	0.368	1.316	1.690	0.402
			Multi-MinMax	0.605	0.560	0.169	0.513	1.521	0.194
		40 x 5	Sum-MinMax	0.736	0.711	0.326	1.091	1.776	0.385
			Multi-MinMax	0.731	0.689	0.345	1.220	3.057	0.380
			Sum-MinMax	0.722	0.714	0.306	0.944	1.570	0.377
	Ellipsoid	40 x 2	Multi-MinMax	0.540	0.524	0.102	0.298	2.009	0.104
			Sum-MinMax	<u>0.740</u>	0.711	<u>0.352</u>	<u>1.228</u>	1.715	<u>0.396</u>
			Multi-MinMax	0.611	0.561	0.171	0.522	<u>1.611</u>	0.199
		40 x 5	Sum-MinMax	0.740	<u>0.714</u>	0.330	1.100	1.799	0.335
			Multi-MinMax	0.730	0.699	0.341	1.220	3.057	0.380
			Sum-MinMax	0.731	0.689	0.343	1.220	3.018	0.380

Table 5.8: Analysis of **GVA gaussian on synthetics images dataset** on CAT2000 and SID4VAM, using AlexNet as base network.

5.4. Conclusion

Dataset	Fusion	Normalization	AUC-Judd	AUC-Borji	CC	NSS	KL↓	SIM
MIT1003	Mult	MinMax	0.796	0.738	0.358	1.099	1.737	0.355
		Energy	0.795	0.737	0.357	1.182	1.797	0.357
		Standarization	0.794	0.737	0.336	1.180	1.797	0.356
	Sum	MinMax	0.795	0.786	0.364	1.186	1.515	0.311
		Energy	0.794	0.785	0.364	1.185	1.516	0.310
		Standarization	0.792	0.785	0.363	1.185	1.516	0.310
Toronto	Mult	MinMax	0.796	0.715	0.479	1.445	2.499	0.439
		Energy	0.795	0.713	0.479	1.444	2.498	0.439
		Standarization	0.795	0.713	0.477	1.444	2.499	0.438
	Sum	MinMax	0.792	0.777	0.465	1.356	1.221	0.393
		Energy	0.791	0.776	0.464	1.356	1.220	0.393
		Standarization	0.791	0.776	0.464	1.355	1.221	0.393
KTH	Mult	MinMax	0.629	0.597	0.274	0.397	1.300	0.511
		Energy	0.628	0.596	0.274	0.395	1.299	0.512
		Standarization	0.627	0.596	0.274	0.395	1.301	0.511
	Sum	MinMax	0.634	0.629	0.323	0.466	0.669	0.559
		Energy	0.633	0.627	0.322	0.464	0.66	0.555
		Standarization	0.632	0.628	0.322	0.464	0.668	0.554
CAT2000	Mult	MinMax	0.813	0.735	0.655	1.688	1.541	0.560
		Energy	0.811	0.733	0.651	1.688	1.540	0.560
		Standarization	0.811	0.733	0.653	1.687	1.541	0.559
	Sum	MinMax	0.789	0.775	0.518	1.321	0.992	0.452
		Energy	0.787	0.777	0.518	1.322	0.993	0.451
		Standarization	0.788	0.776	0.518	1.322	0.992	0.452
SID4VAM	Mult	MinMax	0.747	0.708	0.383	1.377	2.142	0.404
		Energy	0.744	0.707	0.383	1.375	2.142	0.404
		Standarization	0.746	0.707	0.382	1.375	2.143	0.403
	Sum	MinMax	0.744	0.733	0.338	1.043	1.433	0.382
		Energy	0.742	0.733	0.336	1.043	1.475	0.381
		Standarization	0.741	0.734	0.337	1.043	1.474	0.381

Table 5.9: Analysis of adquisition on MIT1003, Toronto, KTH, CAT2000 and SID4VAM, using AlexNet as base-network

Chapter 5. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition

	GVA	Fusion	TORONTO	MIT1003	KTH	CAT2000	SID4VAM
Circular	35 x 2	Multi-MinMax	0.616	0.595	0.521	0.620	0.525
		Sum-MinMax	0.770	0.745	0.600	0.732	0.741
	35 x 5	Multi-MinMax	0.762	0.718	0.578	0.753	0.605
		Sum-MinMax	0.781	0.768	0.609	0.780	0.736
	35 x 14	Multi-MinMax	0.792	0.792	0.632	0.812	0.730
		Sum-MinMax	0.789	0.794	0.635	0.819	0.722
Ellipsoid	35 x 2	Multi-MinMax	0.640	0.651	0.527	0.678	0.540
		Sum-MinMax	0.776	0.758	0.597	0.751	0.740
	35 x 5	Multi-MinMax	0.780	0.724	0.581	0.759	0.611
		Sum-MinMax	0.788	0.771	0.620	0.790	0.740
	35 x 14	Multi-MinMax	0.800	0.799	0.639	0.812	0.730
		Sum-MinMax	0.801	0.800	0.640	0.820	0.730
SCB	-	Multi-MinMax	0.796	0.796	0.628	0.812	0.746
SCB	-	Sum-MinMax	0.793	0.795	0.634	0.787	0.741

Table 5.10: Ablation of fusion and normalization on all saliency datasets. We show results for the AUC-Judd metric (top-1 fusion is **bold**)

5.4. Conclusion


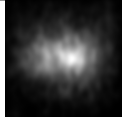
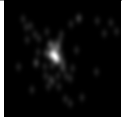
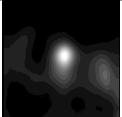
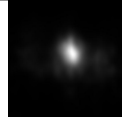
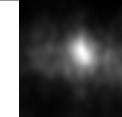



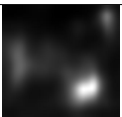
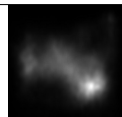
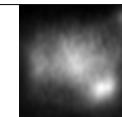

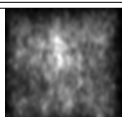
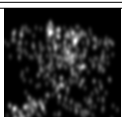
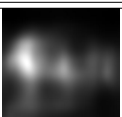
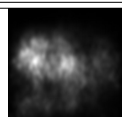

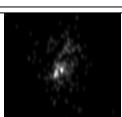
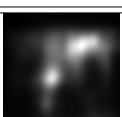
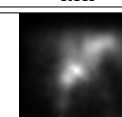
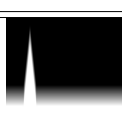
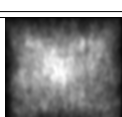
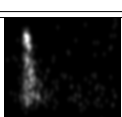
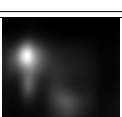
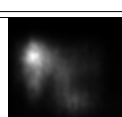
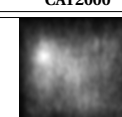
Image	CB	GT	SM	SM+Fusion MULT	SM+Fusion SUM
					
TORONTO					
					
MIT1003					
					
KTH					
					
CAT2000					
					
SID4VAM					

Table 5.11: Qualitative results using human center bias

Chapter 5. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition

Method	AUC	KL ↓	SIM	sAUC	InfoGain
IKN [76]	<u>0.782</u>	<u>1.249</u>	0.366	0.650	-0.024
AIM [18]	0.716	1.612	0.314	0.663	-0.580
SDLF [177]	0.703	1.518	0.304	0.664	-0.398
GBVS [60]	<u>0.803</u>	<u>1.168</u>	0.397	0.632	<u>0.077</u>
DeepGazeII [97]	0.838	1.367	0.325	0.763	-0.200
SAM-ResNet [30]	0.725	2.420	0.516	<u>0.666</u>	-1.555
OpenSALICON [72, 175]	0.771	1.113	0.429	0.716	<u>0.232</u>
SalGAN [137]	<u>0.818</u>	1.272	<u>0.435</u>	<u>0.715</u>	0.392
Our Approach (Step I)	0.731	1.513	0.394	0.589	-0.418
<i>Ground Truth (Humans)</i>	<i>0.954</i>	<i>0.000</i>	<i>1.000</i>	<i>0.902</i>	<i>2.425</i>

Table 5.12: Comparison our saliency output with on standard benchmark methods over synthetic image datasets (Left: Toronto, Right: SID4VAM) for saliency prediction. (Top) Baseline low-level saliency models. (Bottom) State-of-the-art deep saliency models. Best score for each metric is defined as **bold** and TOP-3 scores are underlined.

Method	AUC	KL ↓	SIM	sAUC	InfoGain
IKN [76]	<u>0.678</u>	1.748	0.380	0.608	-0.233
AIM [18]	0.566	14.472	0.224	0.557	-18.181
SDLF [177]	0.607	3.954	0.322	0.596	-3.244
GBVS [60]	<u>0.718</u>	1.363	0.413	0.628	0.331
DeepGazeII [97]	0.610	<u>1.434</u>	0.335	0.571	-0.964
SAM-ResNet [30]	0.673	2.610	<u>0.388</u>	0.600	-1.475
OpenSALICON [72, 175]	0.673	<u>1.549</u>	0.375	<u>0.615</u>	<u>0.052</u>
SalGAN [137]	0.662	2.506	0.373	0.593	-1.350
Our Approach (Step I)	0.721	1.663	<u>0.409</u>	<u>0.627</u>	-0.125
<i>Ground Truth (Humans)</i>	<i>0.882</i>	<i>0.000</i>	<i>1.000</i>	<i>0.860</i>	<i>2.802</i>

Table 5.13: Comparison our saliency output with on standard benchmark methods over synthetic image datasets (Left: Toronto, Right: SID4VAM) for saliency prediction. (Top) Baseline low-level saliency models. (Bottom) State-of-the-art deep saliency models. Best score for each metric is defined as **bold** and TOP-3 scores are underlined.

5.4. Conclusion

Model	TORONTO	MIT1003	KTH	CAT2000	SID4VAM
IKN	0.366	0.290	0.547	0.382	0.380
AIM	0.314	0.251	0.523	0.301	0.224
SDLF	0.304	0.251	0.512	0.309	0.322
GBVS	0.397	0.324	0.563	0.430	0.413
DeepGazell	0.325	0.260	0.549	0.335	0.335
ML-Net	0.489	0.424	0.557	0.375	0.373
SAM-VGG	0.214	0.182	0.354	0.322	0.216
SAM-ResNet	0.516	0.472	0.508	0.456	0.388
SalGAN	0.435	0.435	0.544	0.553	0.373
Best Network	0.439	0.284	0.499	0.406	0.379
Best Network+SCB [×]	0.442	0.299	0.501	0.561	0.388
Best Network+UCBc [×] *	0.447	0.303	0.502	0.537	0.390
Best Network+UCBe ⁺ *	0.449	0.307	0.505	0.544	0.394
Humans (GT)	1.000	1.000	1.000	1.000	1.000

SCB⁺: Supervised (baseline from fixation data), UCB^{×/+}(c/e): Unsupervised with circular (c) or Ellipsoid (e) gaussian center bias.

* Selecting most similar GVA to SCB per dataset

Table 5.14: Benchmark on saliency models. We show results for the **SIM** metrics and state of the art (top-1 model is **bold**)

Chapter 5. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition

Model	TORONTO	MIT1003	KTH	CAT2000	SID4VAM
IKN	0.794	0.760	0.617	0.701	0.686
AIM	0.727	0.706	0.572	0.570	0.570
SDLF	0.714	0.697	0.555	0.573	0.620
GBVS	0.817	0.807	0.649	0.759	0.747
DeepGazeII	0.850	0.849	0.648	0.612	0.612
ML-Net	0.845	0.839	0.658	0.678	0.700
SAM-VGG	0.569	0.559	0.525	0.625	0.537
SAM-ResNet	0.850	0.854	0.660	0.766	0.727
SalGAN	0.821	0.856	0.655	0.751	0.715
Best Network	0.782	0.723	0.615	0.722	0.699
Best Network+SCB [×]	0.810	0.808	0.641	0.820	0.711
Best Network+UCBc ^{×*}	0.812	0.809	0.643	0.819	0.708
Best Network+UCBe ^{+*}	0.813	0.810	0.645	0.822	0.710
Humans (GT)	0.969	0.978	0.902	0.895	0.943

SCB⁺: Supervised (baseline from fixation data), UCB[×]/⁺(c/e): Unsupervised with circular (c) or Ellipsoid (e) gaussian center bias.

* Selecting most similar GVA to SCB per dataset

Table 5.15: Benchmark on saliency models. We show results for the **AUC-Judd** metrics and state of the art (top-1 model is **bold**)

6 Conclusions and Future Work

6.1 Conclusions

This dissertation focused on the role of saliency to improve the classification accuracy of a CNN. Our first approach consisted in adding a saliency branch to an existing CNN architecture which is used to modulate the standard bottom-up visual features from the original image input, acting as an attentional mechanism that guides the feature extraction process. The main aim of the proposed approach was to enable the effective training of a fine-grained recognition model with limited training samples and to improve the performance on the task, thereby alleviating the need to annotate a large dataset. The vast majority of saliency methods are evaluated on their ability to generate saliency maps, and not on their functionality in a complete vision pipeline. Our proposed pipeline allows to evaluate saliency methods for the high-level task of object recognition. We performed extensive experiments on various fine-grained datasets (Flowers, Birds, Cars, and Dogs) under different conditions and show that saliency can considerably improve the network's performance, especially for the case of scarce training data. Furthermore, our experiments showed that saliency methods that obtain improved saliency maps (as measured by traditional saliency benchmarks) also translate to saliency methods that yield improved performance gains when applied in an object recognition pipeline.

In our second approach, we set out to address one of the main disadvantages of the SMIC method. Therefore, we proposed an approach that does not require explicit saliency maps to improve image classification, but these are learned implicitly

during end-to-end image classification task training. We showed that our approach achieves similar results to the case where saliency maps are explicitly provided. We validated our method on several data sets for fine-grained classification tasks (flowers, birds, and cars) and showed that, especially for domains with limited data, the proposed method significantly improves the results.

And finally, we were able to go deeper into the use of saliency in Chapter 5, where we demonstrated that it is possible to automatically generate saliency maps without any truth about the terrain. In our last approach, saliency maps were learned as a side effect of training an object recognition network that is endowed with a saliency branch. Extensive experiments carried out on both real and synthetic saliency datasets demonstrated that our approach is capable of generating accurate saliency maps, achieving competitive results on both synthetic and real datasets when compared to methods that do require ground truth data.

6.1.1 List of Publications

This thesis covers the following publications, in chronological order:

- **Carola Figueroa-Flores**, Abel Gonzalez-Garcia, Joost van de Weijer and Bogdan Raducanu. Saliency for fine-grained object recognition in domain with scarce training data. *Pattern Recognition* ; 94-62-73, 2019 (Journal).
- **Carola Figueroa Flores**, Bogdan Raducanu, David Berga and Joost van de Weijer. Hallucinating saliency maps for fine-grained image classification for limited data domains. In the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2021) *paper accepted as a Full Paper*.
- **Carola Figueroa Flores**, David Berga, Bogdan Raducanu and Joost van de Weijer. Saliency for Free: Saliency Prediction as a Side-Effect of Object Recognition. *Submitted to: Pattern Recognition Journal, 2020 under review*

6.2 Future Work

The purpose of this section is to describe possible future experiments and propose lines of study that were not approached during the development of this thesis.

Thanks to the research efforts of the computer vision community on the ad-

vent of deep neural networks and large annotated datasets, saliency prediction techniques have presented a gaze-assisted attention mechanism for image caption based on human eye fixations (i.e. the static states of gaze upon a specific location). Although this strategy confirms the importance of using eye fixations, it requires gaze information from a human operator. Therefore, it can not be applied on general visual data archives, in which this information is missing. To overcome this limit, Tavakoli et al. [174] presented an image captioning method based on saliency maps, which can be automatically predicted from the input image. Based on this approach and our results obtained in **Chapter 3**, we could propose as future work an approach which incorporates saliency prediction to effectively enhance the quality of image description. This way, we could open an opportunity to extend this work to the problem of "neural image captions", that is, how to provide a textual description for the most salient region(s) of an image.

How quickly can you tell the number of salient objects in an image? As early as the 19th century, it was observed that humans can effortlessly identify the number of objects in the range of 1-4 at a glance [165]. Since then, this phenomenon, later coined by Kaufman et al. [83] as *subitizing*, has been studied and tested in various experimental settings [200]. Inspired by the subitizing phenomenon and our results obtained in our model in **Chapter 4**, we would be interested to study the problem of salient object subitizing (SOS), i.e. predicting the existence and the number of salient objects in an image without using any localization process.

After demonstrating that our model proposed in chapter 5 has achieved good results, we would like to extend it, using for the concept related to "multi-scale saliency". For this, we could introduce contrast-based element saliency at each scale. Finally, a multi-scale saliency integration strategy would be applied to obtain the final saliency map.

Bibliography

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*, 2016.
- [2] Amjad Almahairi, Nicolas Ballas, Tim Cooijmans, Yin Zheng, Hugo Larochelle, and Aaron Courville. Dynamic capacity networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, pages 2549–2558, 2016.
- [3] Roland J. Baddeley and Benjamin W. Tatler. High frequency edges (but not contrast) predict where we fixate: A bayesian system identification analysis. Vision Research, 46(18):2824 – 2833, 2006.
- [4] David Balduzzi, Marcus Frean, Lennox Leary, J.P. Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In Proceeding of International Conference on Machine Learning (ICML), pages 342–350, 2017.
- [5] Moshe Bar and Irving Biederman. Subliminal visual priming. Psychological Science, 9(6):464–468, 1998.

Bibliography

- [6] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Proceedings of Neural Information Processing Systems, pages 585–591, 2001.
- [7] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives, 2014.
- [8] T. Berg, Jiongxin Liu, S. Lee, M. L. Alexander, D. Jacobs, and P. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 2019–2026, 2014.
- [9] David Berga, Xose R. Fdez-Vidal, Xavier Otazu, and Xose M. Pardo. Sid4vam: A benchmark dataset with synthetic images for visual attention modeling. In Proceedings of the IEEE International Conference on Computer Vision, pages 8789–8798, October 2019.
- [10] A. Borji. Boosting bottom-up and top-down visual features for saliency estimation. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pages 438–445, 2012.
- [11] A. Borji, D. N. Sihite, and L. Itti. What/where to look next? modeling top-down visual attention in complex interactive environments. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 44(5):523–538, 2014.
- [12] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1):185–207, jan 2013.
- [13] Ali Borji and Laurent Itti. Cat2000: A large scale fixation dataset for boosting saliency research. In Workshop on “Future of Datasets”, 2015.
- [14] Ali Borji and James Tanner. Reconciling saliency and object center-bias hypotheses in explaining free-viewing fixations. IEEE Transactions on Neural Networks and Learning Systems, 27(6):1214–1226, jun 2016.
- [15] Ali Borji, Hamed R. Tavakoli, Dicky N. Sihite, and Laurent Itti. Analysis of scores, datasets, and models in visual saliency prediction. In Proceedings of the IEEE International Conference on Computer Vision. IEEE, December 2013.
- [16] Dirk Brockmann and Theo Geisel. Are human scanpaths levy flights? 1999.

- [17] N. D. B. Bruce and J. K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. Journal of Vision, 9(3):5–5, mar 2009.
- [18] Neil D. B. Bruce and John K. Tsotsos. Saliency based on information maximization. In Proceedings of the 18th International Conference on Neural Information Processing Systems, NIPS’05, pages 155–162, Cambridge, MA, USA, 2005. MIT Press.
- [19] Neil D.B. Bruce, Calden Wloka, Nick Frosst, Shafin Rahman, and John K. Tsotsos. On computational modeling of visual saliency: Examining what’s right, and what’s left. Vision Research, 116:95–112, nov 2015.
- [20] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark.
- [21] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? arXiv preprint, 2016.
- [22] 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.
- [23] Charles F. Cadieu, Ha Hong, Daniel L. K. Yamins, Nicolas Pinto, Diego Ardila, Ethan A. Solomon, Najib J. Majaj, and James J. DiCarlo. Deep neural networks rival the representation of primate it cortex for core visual object recognition. PLOS Computational Biology, 10(12):1–18, 12 2014.
- [24] S. Cai, W. Zuo, and L. Zhang. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In Proceedings of the IEEE International Conference on Computer Vision, pages 511–520, 2017.
- [25] Chuanbo Chen, He Tang, Zehua Lyu, Hu Liang, Jun Shang, and Mudar Serem. Saliency modeling via outlier detection. Journal of Electronic Imaging, 23(5):053023, 2014.
- [26] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li. Disc: Deep image saliency computing via progressive representation learning. IEEE Transactions on Neural Networks and Learning Systems, 27(6):1135–1149, 2016.
- [27] Y. Chen, Y. Bai, Wei Zhang, and T. Mei. Destruction and construction learning for fine-grained image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5157–5166, 2019.

Bibliography

- [28] François Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1251–1258, 2017.
- [29] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In International Conference on Pattern Recognition (ICPR), 2016.
- [30] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting Human Eye Fixations via an LSTM-based Saliency Attentive Model. IEEE Transactions on Image Processing, 27(10):5142–5154, 2018.
- [31] David Crundall and Geoffrey Underwood. Visual attention while driving: measures of eye movements used in driving research. In Handbook of traffic psychology, pages 137–148. Elsevier, 2011.
- [32] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge J. Belongie. Large scale fine-grained categorization and domain-specific transfer learning. CoRR, abs/1806.06193, 2018.
- [33] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A survey of deep learning and its applications: A new paradigm to machine learning. Archives of Computational Methods in Engineering, pages 1–22, 2019.
- [34] Walter F Dearborn. How people look at pictures: A study of the psychology of perception in art. guy thomas buswell. The Elementary School Journal, 37(1):66–67, 1936.
- [35] Li Deng, Dong Yu, et al. Deep learning: Methods and applications. Foundations and Trends® in Signal Processing, 7(3–4):197–387, 2014.
- [36] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, and J. Jiao. Selective sparse sampling for fine-grained image recognition. In Proceedings of the IEEE International Conference on Computer Vision, pages 6599–6608, 2019.
- [37] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute-guided augmentation. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pages 3328–3336, 2017.
- [38] Ruoyi Du, Dongliang Chang, Ayan Kumar Bhunia, Jiyang Xie, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Fine-grained visual classification via progressive

- multi-granularity training of jigsaw patches. In Proceedings of European Conference on Computer Vision, pages 153–168. Springer, 2020.
- [39] Wolfgang Einhäuser, Merrielle Spain, and Pietro Perona. Objects predict fixations better than early saliency. Journal of vision, 8(14):18–18, 2008.
- [40] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In Proceedings of European Conference on Computer Vision, 2018.
- [41] Ruo Chen Fan, Ming-Ming Cheng, Qibin Hou, Tai-Jiang Mu, Jingdong Wang, and Shi-Min Hu. S4net: Single stage salient-instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6103–6112, 2019.
- [42] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. IEEE Transactions on Pattern Analysis Machine Intelligence, 28(4):594–611, April 2006.
- [43] John M Findlay. Saccadic eye movement programming: Sensory and attentional factors. Psychological research, 73(2):127–135, 2009.
- [44] Burkhard Fischer and Heike Weber. Express saccades and visual attention. Behavioral and Brain Sciences, 16(3):553–567, 1993.
- [45] Ronald A Fisher. The use of multiple measurements in taxonomic problems. Annals of eugenics, 7(2):179–188, 1936.
- [46] Carola Figuera Flores, Abel Gonzalez-Garcia, Joost van de Weijer, and Bogdan Raducanu. Saliency for fine-grained object recognition in domains with scarce training data. Pattern Recognition, 94:62–73, 2019.
- [47] Carola Figuera Flores, Bogdan Raducanu, David Berga, and Joost van de Weijer. Hallucinating saliency maps for fine-grained image classification for limited data domains. arXiv preprint arXiv:2007.12562v1, 2020.
- [48] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11):559–572, 1901.
- [49] J. Fu, H. Zheng, and T. Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pages 4476–4484, July 2017.

Bibliography

- [50] J. Fu, H. Zheng, and T. Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4438–4446, 2017.
- [51] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. CoRR, abs/1511.06062, 2015.
- [52] Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars. Fine-grained categorization by alignments. In Proceedings of the IEEE international conference on computer vision, pages 1713–1720, 2013.
- [53] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. CoRR, abs/1702.08690, 2017.
- [54] Bashir Ghariba, Mohamed S Shehata, and Peter McGuire. Visual saliency prediction based on deep learning. Information, 10(8):257, 2019.
- [55] Bashir Muftah Ghariba, Mohamed S Shehata, and Peter McGuire. A novel fully convolutional network for visual saliency prediction. PeerJ Computer Science, 6:e280, 2020.
- [56] Xavier Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research - Proceedings Track, 9:249–256, 01 2010.
- [57] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT press, 2016.
- [58] Nurit Gronau. Vision at a glance: The role of attention in processing object-to-object categorical relations. Attention, Perception, Psychophysics, 82, 01 2020.
- [59] Sunhyoung Han and Nuno Vasconcelos. Biologically plausible saliency mechanisms improve feedforward object recognition. Vision Research, 50:2295—2307, 2010.
- [60] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems 19, pages 545–552. MIT Press, 2007.

- [61] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In Proceedings of the IEEE International Conference on Computer Vision, pages 3018–3027, 2017.
- [62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
- [63] Shengfeng He, Jianbo Jiao, Xiaodan Zhang, Guoqiang Han, and Rynson W.H. Lau. Delving into salient object subitizing and detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 1059–1067, 2017.
- [64] Xiangteng He and Yuxin Peng. Fine-grained image classification via combining vision and language. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5994–6002, 2017.
- [65] John M. Henderson and Taylor R. Hayes. Meaning guides attention in real-world scene images: Evidence from eye movements and meaning maps. bioRxiv, 2018.
- [66] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural computation, 18(7):1527–1554, 2006.
- [67] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006.
- [68] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 826–834, 2016.
- [69] Grant Van Horn, Oisin Mac Aodha, Yang Song, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist challenge 2017 dataset. CoRR, abs/1707.06642, 2017.
- [70] S. Hou, Y. Feng, and Z. Wang. Vegfru: A domain-specific dataset for fine-grained visual categorization. In Proceedings of the IEEE International Conference on Computer Vision, pages 541–549, 2017.
- [71] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1173–1182, 2016.

Bibliography

- [72] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In IEEE International Conference on Computer Vision, pages 262–270, 2015.
- [73] Laurent Itti. Visual salience. Scholarpedia, 2(9):3327, 2007.
- [74] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. Vision Research, 40(10):1489–1506, 2000.
- [75] Laurent Itti and Christof Koch. Computational modeling of visual attention. Nature reviews. Neuroscience, 2:194–203, 04 2001.
- [76] 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.
- [77] R. Ji, L. Wen, L. Zhang, D. Du, Y. Wu, C. Zhao, X. Liu, and F. Huang. Attention convolutional binary neural tree for fine-grained visual categorization. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 10468–10477, 2020.
- [78] Bowen Jiang, Lihe Zhang, Huchuan Lu, Chuan Yang, and Ming-Hsuan Yang. Saliency detection via absorbing markov chain. In Proceedings of the IEEE International Conference on Computer Vision, pages 1665–1672, 2013.
- [79] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2083–2090, 2013.
- [80] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In Conference on Computer Vision and Pattern Recognition, pages 1072–1080, 2015.
- [81] Tilke Judd, Krista Ehinger, Fredo Durand, and Antonio Torralba. Learning to predict where humans look. In Proceedings of the IEEE International Conference on Computer Vision, pages 2106–2113. IEEE, September 2009.
- [82] Sezer Karaoglu, Ran Tao, Jan C van Gemert, and Theo Gevers. Con-text: Text detection for fine-grained object classification. IEEE transactions on image processing, 26(8):3965–3980, 2017.

- [83] Edna L Kaufman, Miles W Lord, Thomas Whelan Reese, and John Volkmann. The discrimination of visual number. The American journal of psychology, 62(4):498–525, 1949.
- [84] Stephanie J Kayser, Marios G Philiastides, and Christoph Kayser. Sounds facilitate visual motion discrimination via the enhancement of late occipital visual representations. Neuroimage, 148:31–41, 2017.
- [85] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In First Workshop on Fine-Grained Visual Categorization at CVPR, 2011.
- [86] J. Kim, D. Han, Y. Tai, and J. Kim. Salient region detection via high-dimensional color transform. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 883–890, 2014.
- [87] S. Kong and C. Fowlkes. Low-rank bilinear pooling for fine-grained classification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 365–374, 2017.
- [88] Shu Kong and Charless C. Fowlkes. Low-rank bilinear pooling for fine-grained classification. CoRR, abs/1611.05109, 2016.
- [89] Gert Kootstra, Bart de Boer, and Lambert R. B. Schomaker. Predicting eye fixations on complex visual stimuli using local symmetry. Cognitive Computation, 3(1):223–240, jan 2011.
- [90] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In Proceedings of the IEEE International Conference on Computer Vision, pages 1–8, 2013.
- [91] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In IEEE Conference on Computer Vision and Pattern Recognition, pages 5546–5555, 2015.
- [92] Ravikiran Krishnan and Sudeep Sarkar. Conditional distance based matching for one-shot gesture recognition. Pattern Recognition, 48(4):1302 – 1314, 2015.
- [93] 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.

Bibliography

- [94] 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.
- [95] Julia Kucerova and Elena Sikudova. Saliency map augmentation with facial detection. In Proceedings of the 15th Central European seminar on computer graphics, 2011.
- [96] Matthias Kummerer, Thomas SA Wallis, Leon A Gatys, and Matthias Bethge. Understanding low-and high-level contributions to fixation prediction. In Proceedings of the IEEE International Conference on Computer Vision, pages 4789–4798, 2017.
- [97] Matthias Kümmerer, Thomas S. A. Wallis, and Matthias Bethge. Deepgaze ii: Reading fixations from deep features trained on object recognition. arXiv preprint arXiv:1610.01563, 2016.
- [98] Olivier Le Meur and Antoine Coutrot. Introducing context-dependent and spatially-variant viewing biases in saccadic models. Vision research, 121:72–84, 2016.
- [99] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10):1995, 1995.
- [100] R John Leigh and David S Zee. The neurology of eye movements. OUP USA, 2015.
- [101] Olivier LeMeur and Thierry Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. Behavior Research Methods, 45(1):251–266, jul 2012.
- [102] Ang Li, Changyang Li, X. Wang, S. Eberl, D. Feng, and M. Fulham. Automated segmentation of prostate mr images using prior knowledge enhanced random walker. 2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pages 1–7, 2013.
- [103] Aoqi Li, Yingxue Zhang, and Zhenzhong Chen. Scanpath mining of eye movement trajectories for visual attention analysis. In 2017 IEEE International Conference on Multimedia and Expo (ICME), pages 535–540. IEEE, 2017.
- [104] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5455–5463, 2015.

- [105] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 478–487, 2016.
- [106] X. Li, L. Zhao, L. Wei, M. H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deep-saliency: Multi-task deep neural network model for salient object detection. IEEE Transactions on Image Processing, 25(8):3919–3930, 2016.
- [107] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In IEEE Conference on Computer Vision and Pattern Recognition, pages 280–287, 2014.
- [108] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: deep localization, alignment and classification for fine-grained recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1666–1774, 2015.
- [109] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In Proceedings of the IEEE International Conference on Computer Vision, 2015.
- [110] Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pages 9090–9098, 2018.
- [111] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(2):353–367, 2011.
- [112] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, June 2016.
- [113] Manolis Loukidakis, José Cano, and Michael O’Boyle. Accelerating deep neural networks on low power heterogeneous architectures. eprints, 2018.
- [114] David G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal Computer Vision, 60(2):91–110, November 2004.

Bibliography

- [115] Song Lu, Vijay Mahadevan, and Nuno Vasconcelos. Learning optimal seeds for diffusion-based salient object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2790–2797, 2014.
- [116] Ao Luo, Xin Li, Fan Yang, Zhicheng Jiao, Hong Cheng, and Siwei Lyu. Cascade graph neural networks for rgb-d salient object detection. In Proceedings of European Conference on Computer Vision, 2020.
- [117] Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S. Davis, Jun Li, Jian Yang, and Ser-Nam Lim. Cross-x learning for fine-grained visual categorization. In Proceedings of the IEEE International Conference on Computer Vision, pages 8242–8251, 2019.
- [118] A. Mahdi, J. Qin, and G. Crosby. Deepfeat: A bottom-up and top-down saliency model based on deep features of convolutional neural networks. IEEE Transactions on Cognitive and Developmental Systems, 12(1):54–63, 2020.
- [119] Ali Mahdi and Jun Qin. An extensive evaluation of deep features of convolutional neural networks for saliency prediction of human visual attention. Journal of Visual Communication and Image Representation, 65:102662, 2019.
- [120] L. Mai, Y. Niu, and F. Liu. Saliency aggregation: A data-driven approach. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1131–1138, 2013.
- [121] Subhansu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. CoRR, abs/1306.5151, 2013.
- [122] S.K. Mannan, K.H. Ruddock, and D.S. Wooding. Fixation sequences made during visual examination of briefly presented 2d images. Spatial Vision, 11(2):157 – 178, 01 Jan. 1997.
- [123] James A Mazer and Jack L Gallant. Goal-related activity in v4 during free viewing visual search: Evidence for a ventral stream visual salience map. Neuron, 40(6):1241–1250, 2003.
- [124] Rajat Mehrotra, M.A. Ansari, Rajeev Agrawal, and R.S. Anand. A transfer learning approach for ai-based classification of brain tumors. Machine Learning with Applications, 2:100003, 2020.

- [125] Lucia Melloni, Sara van Leeuwen, Arjen Alink, and Notger G. Müller. Interaction between Bottom-up Saliency and Top-down Control: How Saliency Maps Are Created in the Human Brain. Cerebral Cortex, 22(12):2943–2952, 01 2012.
- [126] Jonas Misselhorn, Uwe Frieze, and Andreas K Engel. Frontal and parietal alpha oscillations reflect attentional modulation of cross-modal matching. Scientific reports, 9(1):1–11, 2019.
- [127] Tirin Moore and M Katherine Armstrong. Selective gating of visual signals by microstimulation of frontal cortex. Nature, pages 370–373, 2003.
- [128] T. Munkhdalai and H. Yu. Meta networks. In International Conference on Machine Learning, pages 2554–2563, 2017.
- [129] Francesca Murabito, Concetto Spampinato, Simone Palazzo, Daniela Giordano, Konstantin Pogorelov, and Michael Riegler. Top-down saliency detection driven by visual classification. Computer Vision and Image Understanding, 2018.
- [130] Naila Murray, Maria Vanrell, Xavier Otazu, and C Alejandro Parraga. Saliency estimation using a non-parametric low-level vision model. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 433–440. IEEE, 2011.
- [131] Ryoichi Nakashima, Yu Fang, Yasuhiro Hatori, Akinori Hiratani, Kazumichi Matsumiya, Ichiro Kuriki, and Satoshi Shioiri. Saliency-based gaze prediction based on head direction. Vision Research, 117:59 – 66, 2015.
- [132] T. V. Nguyen, K. Nguyen, and T. Do. Semantic prior analysis for salient object detection. IEEE Transactions on Image Processing, 28(6):3130–3141, 2019.
- [133] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In Sixth Indian Conference on Computer Vision, Graphics & Image Processing, pages 722–729, 2008.
- [134] Tadashi Ogawa and Hidehiko Komatsu. Neuronal dynamics of bottom-up and top-down processes in area v4 of macaque monkeys performing a visual search. Experimental Brain Research, 173(1):1–13, 2006.
- [135] Alice Pailhès, Ronald A. Rensink, and Gustav Kuhn. A psychologically based taxonomy of magicians’ forcing techniques: How magicians influence our choices, and how to use this to study psychological mechanisms. Consciousness and Cognition, 86:103038, 2020.

Bibliography

- [136] Simone Palazzo, Francesco Rundo Rundo, Sebastiano Battiato, Daniela Giordano, and Concetto Spampinato. Visual saliency detection guided by neural signals. In Proceedings of FG, pages 525–531, 2020.
- [137] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E. O’Connor, Jordi Torres, Elisa Sayrol, and Xavier and Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. In arXiv, January 2017.
- [138] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E. O’Connor. Shallow and deep convolutional networks for saliency prediction. In IEEE Conference on Computer Vision and Pattern Recognition, pages 598–606, 2016.
- [139] Derrick Parkhurst, Klinton Law, and Ernst Niebur. Modeling the role of salience in the allocation of overt visual attention. Vision Research, 42(1):107 – 123, 2002.
- [140] Derrick Parkhurst and Ernst Niebur. Scene content selected by active vision. Spatial Vision, 16(2):125 – 154, 01 Jan. 2003.
- [141] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In 2012 IEEE conference on computer vision and pattern recognition, pages 733–740. IEEE, 2012.
- [142] Yongri Piao, Zhengkun Rong, Miao Zhang, Weisong Ren Ren, and Huchuan Lu. A2dele: Adaptive and attentive depth distiller for efficient rgb-d salient object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9060–9069, 2020.
- [143] Radek Ptak. The frontoparietal attention network of the human brain: action, saliency, and a priority map of the environment. The Neuroscientist, 18(5):502–515, 2012.
- [144] Liangqiong Qu, Shengfeng He, Jiawei Zhang, Jiandong Tian, Yandong Tang, and Qingxiong Yang. Rgb-d salient object detection via deep fusion. IEEE Transactions on Image Processing, 26(5):2274–2285, 2017.
- [145] Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä. Segmenting salient objects from images and videos. In European conference on computer vision, pages 366–379. Springer, 2010.

- [146] Subramanian Ramanathan, Harish Katti, Nicu Sebe, Mohan Kankanhalli, and Tat-Seng Chua. An eye fixation database for saliency detection in images. In European Conference on Computer Vision, pages 30–43. Springer, 2010.
- [147] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In International Conference on Learning Representations, 2017.
- [148] Keith Rayner. The 35th sir frederick bartlett lecture: Eye movements and attention in reading, scene perception, and visual search. Quarterly journal of experimental psychology, 62(8):1457–1506, 2009.
- [149] Pamela Reinagel and Anthony M Zador. Natural scene statistics at the centre of gaze. Network: Computation in Neural Systems, 10(4):341–350, 1999. PMID: 10695763.
- [150] Laura Renninger, Preeti Verghese, and James Coughlan. Where to look next? eye movements reduce local uncertainty. Journal of vision, 7:6, 02 2007.
- [151] Eva-Maria Reuter, Welber Marinovic, Timothy N Welsh, and Timothy J Carroll. Increased preparation time reduces, but does not abolish, action history bias of saccadic eye movements. Journal of neurophysiology, 121(4):1478–1490, 2019.
- [152] John H. Reynolds and Robert Desimone. Interacting roles of attention and visual salience in v4. Neuron, 37(5):853 – 863, 2003.
- [153] Nicolas Riche and Matei Mancas. Bottom-up saliency models for still images: A practical review. In From Human Attention to Computational Attention, pages 141–175. Springer New York, 2016.
- [154] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 234–241. Springer, 2015.
- [155] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290(22):2323–2326, 2000.
- [156] K.H. Ruddock, D.S. Wooding, and S. Mannan. Automatic control of saccadic eye movements made in visual inspection of briefly presented 2-d images. Spatial Vision, 9(3):363 – 386, 01 Jan. 1995.

Bibliography

- [157] K.H. Ruddock, D.S. Wooding, and S.K. Mannan. The relationship between the locations of spatial features and those of fixations made during visual examination of briefly presented images. Spatial Vision, 10(3):165 – 188, 01 Jan. 1996.
- [158] 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.
- [159] Bryan Russell, Antonio Torralba, Kevin Murphy, and William Freeman. Labelme: A database and web-based tool for image annotation. International Journal of Computer Vision, 77, 05 2008.
- [160] Alexander C Schütz, Doris I Braun, and Karl R Gegenfurtner. Eye movements and perception: A selective review. Journal of vision, 11(5):9–9, 2011.
- [161] Zahra Sadat Shariatmadar and Karim Faez. Visual saliency detection via integrating bottom-up and top-down information. Optik, 178:1195 – 1207, 2019.
- [162] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, 2015.
- [163] Avishek Siris, Jianbo Jiao, Gary K.L. Tam, Xianghua Xie, and Rynson W.H. Lau. Inferring attention shift ranks of objects for image saliency. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 12133–12143, 2020.
- [164] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems, pages 4077–4087, 2017.
- [165] Jevons Stanley. The power of numerical discrimination. Nature, 3:367–367, 1871.
- [166] H. Strasburger, I. Rentschler, and M. Juttner. Peripheral vision and pattern recognition: A review. Journal of Vision, 11(5):13–13, dec 2011.
- [167] J. Sun, H. Lu, and X. Liu. Saliency region detection based on markov absorption probabilities. IEEE Transactions on Image Processing, 24(5):1639–1649, 2015.

- [168] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-attention multi-class constraint for fine-grained image recognition. In Proceedings of European Conference on Computer Vision, pages 834–850, 2018.
- [169] Yutong Sun, Mohit Prabhushankar, and Ghassan AlRegib. Implicit saliency in deep neural networks. In Proceedings of ICIP, pages 2915–2919, 2020.
- [170] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2818–2826, 2016.
- [171] Benjamin W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. Journal of Vision, 7(14):4, nov 2007.
- [172] Benjamin W. Tatler, Roland J. Baddeley, and Iain D. Gilchrist. Visual correlates of fixation selection: effects of scale and time. Vision Research, 45(5):643 – 659, 2005.
- [173] Benjamin W Tatler and Benjamin T Vincent. The prominence of behavioural biases in eye guidance. Visual Cognition, 17(6-7):1029–1054, 2009.
- [174] Hamed R. Tavakoli, Ali Borji, Jorma Laaksonen, and Esa Rahtu. Exploiting inter-image similarity and ensemble of extreme learners for fixation prediction using deep features. arXiv e-prints, page arXiv:1610.06449, October 2016.
- [175] Christopher Lee Thomas. Opensalicon: An open source implementation of the salicon saliency model. Technical Report TR-2016-02, University of Pittsburgh, 2016.
- [176] Na Tong, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Salient object detection via bootstrap learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1884–1892, 2015.
- [177] Antonio Torralba, Aude Oliva, Monica S. Castelhana, 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):766–786, 2006.
- [178] Anne Treisman and Janet Souther. Search asymmetry: A diagnostic for preattentive processing of separable features. Journal of Experimental Psychology: General, 114(3):285–310, 1985.

Bibliography

- [179] Galina Tremper and Anette Frank. A discriminative analysis of fine-grained semantic relations including presupposition: Annotation and classification. Dialogue & Discourse, 4(2):282–322, 2013.
- [180] Stefan Treue. Visual attention: the where, what, how and why of saliency. Current Opinion in Neurobiology, 13(4):428 – 432, 2003.
- [181] John K. Tsotsos, Scan M. Culhane, Winky Yan Kei Wai, Yuzhong Lai, Neal Davis, and Fernando Nuflo. Modeling visual attention via selective tuning. Artificial Intelligence, 78(1-2):507–545, oct 1995.
- [182] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 595–604, 2015.
- [183] Daan R van Renswoude, Linda van den Berg, Maartje EJ Raijmakers, and Ingmar Visser. Infants’ center bias in free viewing of real-world scenes. Vision research, 154:44–53, 2019.
- [184] Eleonora Vig, Michael Dorr, and David Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2798–2805, 2014.
- [185] Benjamin T. Vincent and Benjamin W. Tatler. Systematic tendencies in scene viewing, 2008.
- [186] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In Advances in Neural Information Processing Systems, 2017.
- [187] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [188] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3156–3164, 2017.
- [189] L. Wang, H. Lu, X. Ruan, and M. Yang. Deep networks for saliency detection via local estimation and global search. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3183–3192, 2015.

- [190] Wenguan Wang and Jianbing Shen. Deep visual attention prediction. IEEE Transactions on Image Processing, 27(5):2368–2378, 2017.
- [191] Wenguan Wang, Jianbing Shen, Ming-Ming Cheng, and Ling Shao. An iterative and cooperative top-down and bottom-up inference network for salient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2019.
- [192] Yawing Wang, Vlad I. Morariu, and Larry S. Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4148–4157, 2018.
- [193] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Barath Hariharan. Low-shot learning from imaginary data. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7278–7286, 2018.
- [194] Z. Wang, S. Wang, S. Yang, H. Li, J. Li, and Z. Li. Weakly supervised fine-grained image classification via gaussian mixture model oriented discriminative learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9749–9758, 2020.
- [195] Xiu-Shen Wei, Quan Cui, Lei Yang, Peng Wang, and Lingqiao Liu. RPC: A large-scale retail product checkout dataset. CoRR, abs/1901.07249, 2019.
- [196] Xiu-Shen Wei, Jianxin Wu, and Quan Cui. Deep learning for fine-grained image analysis: A survey, 2019.
- [197] Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, and Chunhua Shen. Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. Pattern Recognition, 76:704 – 714, 2018.
- [198] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. Geodesic saliency using background priors. In European conference on computer vision, pages 29–42. Springer, 2012.
- [199] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.
- [200] John Whalen, C.R. Gallistel, and Rochel Gelman. Nonverbal counting in humans: The psychophysics of number representation. Psychological Science, 10(2):130–137, 1999.

Bibliography

- [201] Yu-Huan Wu, Yun Liu, Le Zhang, Wang Gao, and Ming-Ming Cheng. Regularized densely-connected pyramid network for salient instance segmentation. arXiv preprint arXiv:2008.12416, 2020.
- [202] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 842–850, 2015.
- [203] Guo-Sen Xie, Xu-Yao Zhang, Wenhan Yang, Mingliang Xu, Shuicheng Yan, and Cheng-Lin Liu. Lg-cnn: From local parts to global discrimination for fine-grained recognition. Pattern Recognition, 71:118–131, 2017.
- [204] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In Proceedings of the IEEE international conference on computer vision, pages 1395–1403, 2015.
- [205] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In International Conference on Machine Learning, pages 2048–2057, 2015.
- [206] Mai Xu, Lai Jiang, Zhaoting Ye, and Zulin Wang. Bottom-up saliency detection with sparse representation of learnt texture atoms. Pattern Recognition, 60:348 – 360, 2016.
- [207] Yingyue Xu, D. Xu, Xiaopeng Hong, Wanli Ouyang, Rongrong Ji, Min Xu, and Guoying Zhao. Structured modeling of joint deep feature and prediction refinement for salient object detection. Proceedings of the IEEE International Conference on Computer Vision, pages 3788–3797, 2019.
- [208] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3166—3173, 2013.
- [209] Jimei Yang and Ming-Hsuan Yang. Top-down visual saliency via joint crf and dictionary learning. IEEE Trans. on PAMI, 39(3):576–588, 2017.
- [210] Fisher Yu, Dequan Wang, and Trevor Darrell. Deep layer aggregation. CoRR, abs/1707.06484, 2017.

- [211] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In International Conference on Learning Representations, 2017.
- [212] Gregory J Zelinsky. Tam: Explaining off-object fixations and central fixation tendencies as effects of population averaging during search. Visual cognition, 20(4-5):515–545, 2012.
- [213] Chunjie Zhang, Wei Xiong, Jing Liu, Yifan Zhang, Chao Liang, and Qingming Huang. Fine-grained image classification using color exemplar classifiers. In Pacific-Rim Conference on Multimedia, pages 327–336. Springer, 2013.
- [214] Han Zhang, Tao Xu, Mohamed Elhoseiny, Xiaolei Huang, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1143–1152, 2016.
- [215] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2770–2779, 2019.
- [216] Jianming Zhang, Shugao Mai, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomír Mech. Salient object subitizing. Int'l. Journal of Computer Vision, 124:169–186, 2017.
- [217] Jianming Zhang and Stan Sclaroff. Saliency detection: A boolean map approach. In IEEE International Conference on Computer Vision, pages 153–160, 2013.
- [218] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Unconstrained salient object detection via proposal subset optimization. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5733–5742, 2016.
- [219] Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9029–9038, 2018.
- [220] Kaihua Zhang, Lei Zhang, Kin-Man Lam, and David Zhang. A level set approach to image segmentation with intensity inhomogeneity. IEEE Transactions on Cybernetics, 46(2):546–557, 2016.

Bibliography

- [221] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In European Conference on Computer Vision, pages 834–849, 2014.
- [222] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 714–722, 2018.
- [223] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. Picking deep filter responses for fine-grained image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1134–1142, 2016.
- [224] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Ju-Feng Yang, and Ming-Ming Cheng. Egnnet: Edge guidance network for salient object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 8779–8788, 2019.
- [225] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1265–1274, 2015.
- [226] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In Proceedings of European Conference on Computer Vision, 2020.
- [227] H. Zheng, J. Fu, T. Mei, and J. Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In Proceedings of the IEEE International Conference on Computer Vision, pages 5219–5227, Oct 2017.
- [228] Guoqiang Zhong, Li-Na Wang, Xiao Ling, and Junyu Dong. An overview on data representation learning: From traditional feature learning to recent deep learning. The Journal of Finance and Data Science, 2(4):265 – 278, 2016.
- [229] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 2814–2821, 2014.