Hierarchical discriminant saliency network for object recognition

Permalink
https://escholarship.org/uc/item/7h8858tb

Author
Han, Sunhyoung

Publication Date
2011

Peer reviewed|Thesis/dissertation
Hierarchical Discriminant Saliency Network for Object Recognition

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

by

Sunhyoung Han

Committee in charge:

Professor Nuno Vasconcelos, Chair
Professor Garrison W. Cottrell
Professor Kenneth Kreutz-Delgado
Professor Truong Nguyen
Professor Virginia de Sa

2011
Copyright

Sunhyoung Han, 2011

All rights reserved.
The dissertation of Sunhyoung Han is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

________________________________________

________________________________________

________________________________________

________________________________________

________________________________________

Chair

University of California, San Diego

2011
DEDICATION

I dedicate this thesis to my husband, Sungwoo Kwon, for his unconditional support and my lovely daughter, Sua.
TABLE OF CONTENTS

Signature Page ................................................................. iii
Dedication ........................................................................ iv
Table of Contents ............................................................... v
List of Figures ................................................................. ix
List of Tables ................................................................... xiv
Acknowledgements ............................................................. xvi
Vita .................................................................................. xix
Publications ...................................................................... xix
Abstract of the Dissertation ................................................ xxii

Chapter I  Introduction ......................................................... 1
  I.A. Hierarchical Discriminant Saliency Network for Object Recognition 2
  I.B. Contributions of the thesis ............................................. 6
    I.B.1. Top-down discriminant saliency ............................... 6
    I.B.2. Biological Plausibility of top-down discriminant saliency ... 7
    I.B.3. Discriminant saliency network ................................. 8
    I.B.4. Application: Amorphous object detection in the wild ...... 10
    I.B.5. Application: Automatic region-of-interest detection for image compression .................................................. 11
  I.C. Organization of the thesis ............................................. 13

Chapter II  Top-down discriminant saliency .......................... 14
  II.A. Saliency principles ................................................... 15
    II.A.1. Discriminant saliency principles ............................ 15
    II.A.2. The minimum Bayes error principle ........................ 16
    II.A.3. Relaxation and the principle of minimum uncertainty ... 18
    II.A.4. Infomax and Barlow’s principle of suspicious coincidences . 18
  II.B. Learning salient features ........................................... 20
    II.B.1. Image statistics and computational efficiency 20
    II.B.2. Computational parsimony and suspicious coincidences 22
    II.B.3. The generalized Gaussian distribution ..................... 23
  II.C. Discriminant saliency ................................................ 25
    II.C.1. Feature selection ................................................. 25
    II.C.2. Feature learning ............................................... 26
    II.C.3. Saliency detection .............................................. 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>II.C.4</td>
<td>Discriminant saliency as a focus-of-attention mechanism</td>
<td>31</td>
</tr>
<tr>
<td>II.D.</td>
<td>Discriminant saliency detection</td>
<td>33</td>
</tr>
<tr>
<td>II.D.1</td>
<td>Discriminant saliency map</td>
<td>33</td>
</tr>
<tr>
<td>II.E.</td>
<td>Object recognition with discriminant saliency</td>
<td>35</td>
</tr>
<tr>
<td>II.E.1</td>
<td>Candidate features for discriminant saliency</td>
<td>35</td>
</tr>
<tr>
<td>II.E.2</td>
<td>Determining the number of salient features</td>
<td>36</td>
</tr>
<tr>
<td>II.F.</td>
<td>Experimental evaluation</td>
<td>36</td>
</tr>
<tr>
<td>II.F.1</td>
<td>Experimental setup</td>
<td>37</td>
</tr>
<tr>
<td>II.F.2</td>
<td>Object localization</td>
<td>40</td>
</tr>
<tr>
<td>II.F.3</td>
<td>Image classification</td>
<td>46</td>
</tr>
<tr>
<td>II.G.</td>
<td>Conclusion</td>
<td>50</td>
</tr>
<tr>
<td>II.H.</td>
<td>Acknowledgments</td>
<td>51</td>
</tr>
<tr>
<td>III.A.</td>
<td>Introduction</td>
<td>53</td>
</tr>
<tr>
<td>III.B.</td>
<td>Natural image statistics</td>
<td>55</td>
</tr>
<tr>
<td>III.C.</td>
<td>Statistical inference</td>
<td>56</td>
</tr>
<tr>
<td>III.D.</td>
<td>Inference under the GGD</td>
<td>65</td>
</tr>
<tr>
<td>III.E.</td>
<td>Biological plausibility</td>
<td>67</td>
</tr>
<tr>
<td>III.F.</td>
<td>Saliency</td>
<td>70</td>
</tr>
<tr>
<td>III.G.</td>
<td>Results</td>
<td>72</td>
</tr>
<tr>
<td>III.G.1</td>
<td>Experimental set up</td>
<td>73</td>
</tr>
<tr>
<td>III.G.2</td>
<td>Synthetic problem</td>
<td>74</td>
</tr>
<tr>
<td>III.G.3</td>
<td>Caltech101 experiments</td>
<td>77</td>
</tr>
<tr>
<td>III.G.4</td>
<td>Comparison to state-of-the-art results</td>
<td>79</td>
</tr>
<tr>
<td>III.H.</td>
<td>Discussion and Conclusion</td>
<td>82</td>
</tr>
<tr>
<td>III.I.</td>
<td>Acknowledgments</td>
<td>86</td>
</tr>
<tr>
<td>IV.A.</td>
<td>Introduction</td>
<td>88</td>
</tr>
<tr>
<td>IV.B.</td>
<td>Multi-layer discriminant saliency networks</td>
<td>92</td>
</tr>
<tr>
<td>IV.B.1</td>
<td>Hierarchical recognition networks</td>
<td>92</td>
</tr>
<tr>
<td>IV.B.2</td>
<td>HDSN architecture</td>
<td>94</td>
</tr>
<tr>
<td>IV.B.3</td>
<td>Visual representation</td>
<td>95</td>
</tr>
<tr>
<td>IV.B.4</td>
<td>Network training</td>
<td>96</td>
</tr>
<tr>
<td>IV.C.</td>
<td>Relationships to previous models</td>
<td>97</td>
</tr>
<tr>
<td>IV.C.1</td>
<td>Saliency models</td>
<td>97</td>
</tr>
<tr>
<td>IV.C.2</td>
<td>HDSN vs. saliency models</td>
<td>98</td>
</tr>
<tr>
<td>IV.C.3</td>
<td>Biological models</td>
<td>99</td>
</tr>
<tr>
<td>IV.C.4</td>
<td>HDSN vs. biological models</td>
<td>101</td>
</tr>
<tr>
<td>IV.C.5</td>
<td>Popular non-biological models</td>
<td>104</td>
</tr>
<tr>
<td>IV.C.6</td>
<td>HDSN vs. computer vision models</td>
<td>111</td>
</tr>
<tr>
<td>IV.D.</td>
<td>Distinguishing characteristics of the HDSN architecture</td>
<td>112</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure II.1 Feature saliency depends on the viewing context. White color is salient feature in the second image but it is not in the salient in the first image. .......................... 17

Figure II.2 Illustration of the conditional marginal distributions (GGDs) for the responses of a feature, when (a) it is present (strong responses) in the object class \( Y = 1 \) but absent (weak responses) in the null hypothesis \( Y = 0 \), or (b) vice versa. Note that the absence of a feature always leads to narrower GGDs than the presence of the feature. ................. 26

Figure II.3 Distribution example showing maximum discriminacy with (a) original basis and (b) projection to rotated basis .... 28

Figure II.4 Implementation of the discriminant saliency detector. Input image is convolved with features and leads to saliency for each feature and summed up to make saliency map. .... 35

Figure II.5 Classification accuracy vs number of features used by the DSD for (a) cars, (b) dogs and (c) motorbikes. ............... 36

Figure II.6 Some of the basis functions in the DCT feature set. Many of the DCT basis functions can be interpreted as detectors of perceptually relevant image attributes, including edges, corners, t-junctions, and spots. ................................. 38

Figure II.7 Precision-recall of various saliency detectors on 8 object classes in PASCAL2006. .................................. 44

Figure II.8 Saliency maps produced by DSD for various images and objects of interest. Bright pixels flag the most salient image locations. ......................................................... 46

Figure II.9 Salient locations at 40% recall. Top to bottom: original images (objects marked by their bounding boxes), salient locations pruned by DSD, DVW, LSVM, and pLSA. Each circle represents the location and size of a salient point. White (black) indicates that the salient point falls inside (outside) the ground truth bounding box for the object. ... 47

Figure II.10 Salient locations at 40% recall. Top to bottom: original images (objects marked by their bounding boxes), salient locations pruned by DSD, DVW, LSVM, and pLSA. Each circle represents the location and size of a salient point. White (black) indicates that the salient point falls inside (outside) the ground truth bounding box for the object. ... 48

Figure III.1 Histogram of responses of a set of Gabor filters to a collection of natural images, and its MAP fit by the GGD model with \( \beta = .5 \). .......................... 56
Figure III.2 Object recognition with the LLR measure. The learning stage is shown at the top of the figure. Gabor filtering is applied to examples of the target and null class. In this example, the target is the class of airplane objects. The probability distributions of the filter responses are then modelled with the GGD distribution. This enables the detection of objects from the target class in previously unseen images, as shown at the bottom. Given the filter responses to an unseen image, and the GGD estimates learned during training, the LLR is computed at each location of the visual field.

Figure III.3 a) an image, b) magnitude of Gabor responses, c) NLL, d) LLR, e) target posterior probability, f) information, g) LLRC(x), h) IC(x). The bars on the side of each image show the range of values corresponding to the pixel amplitudes.

Figure III.4 a) an image, b) magnitude of Gabor responses, c) NLL, d) LLR, e) target posterior probability, f) information, g) LLRC(x), h) IC(x).

Figure III.5 Non linear transformations of the posterior target probability that produce the information $I(Y; X = x)$ (left) and the confidence measures LLRC(x) (center) and IC(x) (right).

Figure III.6 The NLL is computed by a simple cell that normalizes a feature response $x$ by the responses of its spatially neighboring units.

Figure III.7 A LLR unit divisively normalizes a feature response $x$ differentially, using the outputs of two units that estimate GGD parameters under the target and null hypothesis. With the inclusion of the output non-linearity $\sigma(\cdot)$, this unit computes posterior target probabilities.

Figure III.8 A complex unit pools the responses of simple units within some region $\mathcal{R}$, after passing them through a non-linearity.

Figure III.9 Detection of underlined characters. Top row: training examples from target and non-target class. Bottom rows: examples of test stimuli from the target and non-target class, and layer 1 responses from the three networks considered.

Figure III.10 Top: most discriminant filter (the 4 orientation channels are shown) of the second network layer, for HMAX (left) and HMAX+EC (right). By most discriminant it is meant that this is the filter given larger weight by the linear SVM classifier at the network output. Bottom: example output of the simple cells in layer 2, to target and non-target stimuli.

Figure III.11 Recognition rates on Caltech101, using 30 training examples per class.
Figure III.12 An image from the “accordion” class, and corresponding saliency outputs for Gabor channels of four orientations. Top row: magnitude of the Gabor responses. Center: saliency maps produced by NLL units. Bottom: saliency maps of LLR units. 84

Figure IV.1 Hierarchical feedforward recognition architecture, with two layers and four orientation channels. 93

Figure IV.2 DSN computations. a) S-unit: the LLR is computed by divisive normalizing a feature response $x$ differentially, using the outputs of two units that estimate GGD parameters, under the target and null hypothesis. The output non-linearity $\sigma(\cdot)$ transforms the LLR into a posterior target probability. b) C-unit cell computations: responses of afferent simple units are pooled within some region $\mathcal{R}$, after passing them through a non-linearity. 98

Figure IV.3 Most discriminant patch filter for various object classes from Caltech101. 99

Figure IV.4 Operation sequences per layer of the convolutional network (top) and the HDSN (bottom). The most significant difference is the replacement of half-wave rectification by a saliency-based rectifier, tuned per feature. 103

Figure IV.5 Canonical architecture implemented by various popular object recognition methods. 105

Figure IV.6 half-wave (left) and saliency-based (right) rectifiers. The latter varies significantly with the relative values of $\alpha_1$ and $\alpha_0$. 113

Figure IV.7 Hierarchical feedforward recognition architecture, with four orientation channels. 116

Figure IV.8 Classification performance comparison on ALOI database as a function of training size for various methods. 128

Figure IV.9 Examples of object categories and its saliency map. 25 saliency maps in the bottom show saliency for 25 target classes starting top left to bottom right object. 129

Figure IV.10 Object localization result averaged over 1000 targets. (a) precision-recall curve (b) ROC curve. 130

Figure V.1 Example images from the PandaCam dataset. Note the high variability of view points, illumination, and object pose. 137

Figure V.2 Characteristic function of (a) discriminant saliency and (b) entropy measure. 146
Figure V.3 Architecture of the proposed template-based saliency detector. DCT features are used in the first network layer, and saliency templates in the second. The saliency computation is repeated at multiple scales. 148

Figure V.4 Images from the PandaCam dataset (top row), and saliency maps produced with saliency templates (second row), SIFT templates (third), and discriminant visual words (fourth). 150

Figure V.5 Precision recall curves for object localization. We compare localization performance of template saliency, SIFT saliency, and discriminant visual word. 152

Figure V.6 Examples of detections by the template-based saliency detector. White windows indicate ground truth, blue detections and red false-positives. 153

Figure V.7 Curves of detection rate vs false positives per image. Performance is compared with state-of-the-art computer vision algorithms. 155

Figure VI.1 Example of image which may contain different region of interest. This image contains multiple objects such as big ben, bus, people, car, etc. 158

Figure VI.2 Salient point detection. Image is decomposed into multiple scale layer followed by convolution with feature and finally generates saliency table by applying winner take all network. 163

Figure VI.3 (a) Saliency detection result, (d) Init saliency points overlapped over 40 samples, (b) Overlapped points classified as ”object”, (c) Overlapped points classified as ”noise”, (e) Visualized variance, Σ₁, and (f) Visualized variance, Σ₀. 171

Figure VI.4 Saliency points classification results for some of training examples. White circle belongs more to ”object” class and black one belongs more to ”noise” class. 171

Figure VI.5 (a) Original image, (b) Visualized saliency map, (c) Probability map, P₁, considering the configuration, (d) Final ROI mask for the object of ‘street sign’. 177

Figure VI.6 Examples showing the process to find ROI mask, row 1 is for the object, “Big Ben”, row 2 for “Eiffel Tower”, row 3 for “Statue of Liberty”, and row 4 for “Tower of Pisa”. The first column shows the given test images, the second one shows visualised saliency lists, the third one clustered saliency points, the fourth one gray level of ROI mask, and the fifth column shows the binary ROI masks for given object. 178

Figure VI.7 First row is the result with UIUC car side database and the second row is the result with CalTech face database. (a,d) 1-precision vs Recall curve for UIUC database (b,e) ROC area curve (c,f) PSNR improvements by using ROI coding. 179
Figure VI.8  Automatic image sorting process. Among downloaded images using google search engine, good example images are selected by sorting with saliency value. . . . . . . . . . . . . . 180
Figure VI.9  (a) Comparison of scores for each category (b) ROC curves (c) True Positive rate according to number of images . . . . 182
Figure VI.10 Top row: clean training examples given with data set bottom row: selected training set from downloaded images through web search for the object ”car” . . . . . . . . . . . . . . . . . . . 183
Figure VI.11 Example of typical multi object detection problem in a scene. Gaussian window can be overlapped between objects. . . . . 184
Figure VI.12 First row shows images which are coded with normal jpeg and the bottom row shows the corresponding images coded with detected ROI mask. BPP is 0.2; 23.4Kbits for each example in ”face” class and BPP is 0.6; 14.04Kbits for each example in ”car side” class . . . . . . . . . . . . . . . . . . . . . . . 185
Figure VI.13 Coding examples under the same BPP (a), (d), and (g) are the original images, (b), (e), and (h) are the normal JPEG case, and (c), (f), and (i) is ROI JPEG using the detected ROI mask . . . . . . . . . . . . . . . . . . . . . . . . 187
Figure VI.14 Coding examples under the same BPP. The first column is the images coded with normal JPEG, and the second column is the images of ROI JPEG using the detected ROI mask . . . . . . . . . . . . . . . . . . . . . . . . 188
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table II.1</th>
<th>Decision rules, confidence measures, and feature selection costs with different principles.</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table II.2</td>
<td>Number of training and testing images in 10 object classes PASCAL dataset.</td>
<td>37</td>
</tr>
<tr>
<td>Table II.3</td>
<td>Image classification accuracy (%) of Bag-of-Feature Spatial Pyramid (BFSP) classifiers for PASCAL2006.</td>
<td>40</td>
</tr>
<tr>
<td>Table II.4</td>
<td>Average precision-recall for various detectors. DSD-BU achieves the best performance among others.</td>
<td>45</td>
</tr>
<tr>
<td>Table II.5</td>
<td>Image classification accuracy (%) of Bag-of-Features Spatial Pyramid (BFSP) classifiers for PASCAL2006.</td>
<td>50</td>
</tr>
<tr>
<td>Table III.1</td>
<td>Operations of statistical inference under the GGD model. $\psi(x)$ is defined as $\psi(x) = \frac{1}{2} \log \frac{x}{1-x}$.</td>
<td>58</td>
</tr>
<tr>
<td>Table III.2</td>
<td>Recognition rates on Caltech101, using 30 training examples per class. All abbreviations are the same as in Table III.1. Furthermore, EC means expected confidence, ELLR expected LLR, ENLL expected NLL, RFD rare feature detection.</td>
<td>78</td>
</tr>
<tr>
<td>Table III.3</td>
<td>Multiclass classification results for 101 categories. The table shows the improvement by changing/adding components to base model.</td>
<td>80</td>
</tr>
<tr>
<td>Table IV.1</td>
<td>Comparison of comparable units in various architectures with HDSN: first layer.</td>
<td>108</td>
</tr>
<tr>
<td>Table IV.2</td>
<td>Comparison of comparable units in various architectures with HDSN: second layer.</td>
<td>108</td>
</tr>
<tr>
<td>Table IV.3</td>
<td>DiscSalNet vs BoF with SPK case. This table shows comparison between general DiscSalNet architecture with SIFT + BoF + spatial pyramid case.</td>
<td>119</td>
</tr>
<tr>
<td>Table IV.4</td>
<td>First layer unit comparison. This experiments are done keeping the second layer operation the same and compare results only on the first layer unit.</td>
<td>122</td>
</tr>
<tr>
<td>Table IV.5</td>
<td>Second layer unit comparison. This experiments are done keeping the first layer operation the same and compare results only on the second layer unit.</td>
<td>122</td>
</tr>
<tr>
<td>Table IV.6</td>
<td>Comparison to state-of-the-art approaches in computer vision with number of layer 2 units used.</td>
<td>125</td>
</tr>
<tr>
<td>Table IV.7</td>
<td>Percentage of scenes where the target object is located within the first saliency.</td>
<td>130</td>
</tr>
<tr>
<td>Table V.1</td>
<td>Edgeness statistics for object and background in the PASCAL VOC and PandaCam datasets.</td>
<td>140</td>
</tr>
</tbody>
</table>
Table VI.1  Bits to achieve the same PSNR for two compression methods, ROI vs normal coding. . . . . . . . . . . . . . . . . . . . 189
ACKNOWLEDGEMENTS

Through only my name appears on the cover of this dissertation, a great many people have contributed to its production. I owe my gratitude to all those people who helped and supported me during my doctoral study.

My deepest gratitude is to my advisor, Professor Nuno Vasconcelos. I have been amazingly fortunate to have an advisor who is full of academic enthusiasm. With his enthusiasm, his inspiration, and his great insights, he helped me to become deep thinker. Throughout my thesis-writing period, he provided sound advice, good teaching, and lots of good ideas. I would have been lost without him.

I am also very grateful for having an exceptional doctoral committee and wish to thank Professor Kenneth Kreutz-Delgado, Professor Nguyen Truong, Professor Garrison W. Cottrell, and Professor Virginia de Sa for their valuable input, discussions and accessibility.

I am indebted to my many colleagues of SVCL at UCSD for providing a stimulating environment in which to learn and grow. I am especially grateful to Dr. Dashan Gao for all the mentoring during my early doctoral study. I would like to thank Dr. Antoni B. Chan, Dr. Hamed Masnadi-Shirazi, Nikhil Rasiwasia, Vijay Mahadevan, Kritiika Muralidharan, Mohammad Saberian, Jose Maria Costa Pereira, Weixin Li, and Mandar Dixit for their continuous support, friendship, and the assistance in the past several years.

During my Ph.D study in UCSD, I got to know many friends in San Diego and enjoy their company in many ways. I especially like to thank Dr. Jaewook Shim, Dr. Jiho Park, Dr. Sunghoon Park, Dr. Youngwan Kim for their friendship. They have helped me going through tough times and make my life for the last six years bearable.

I wish to thank my parents, parents-in-law for giving loving and supporting environment for me. I also thank to my sisters, brother, sister-in-law, brother-in-laws in Korea for believing in me. I am especially thankful to my sister Hyejung for helping me take care of my daughter during my study.
Lastly and most importantly, I owe my loving thanks to my husband, Sungwoo Kwon, for the unconditional and endless support to my study. His love and support is the very foundation of my doctoral study and this thesis. I am also grateful to my daughter, Sua, for being there and understanding busy mom. She always has been and will be the source of my comfort.

The text of Chapter II, in full, is based on the material as it appears in: Dashan Gao, Sunhyoung Han and Nuno Vasconcelos, “Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition”, published in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009 and Sunhyoung Han and Nuno Vasconcelos, “Complex discriminant features for object classification”, published in *International Conference on Image Processing*, 2008. The dissertation author was a primary researcher and an author of the cited material.

The text of Chapter III, in full, is based on the material as it appears in: Sunhyoung Han and Nuno Vasconcelos, “Biologically Plausible Saliency Mechanisms Improve Feedforward Object Recognition” , published in *Vision Research*, 2010. The dissertation author was a primary researcher and an author of the cited material.

The text of Chapter IV, in full, is based on the material as it appears in: Sunhyoung Han and Nuno Vasconcelos, “Hierarchical Discriminant Saliency Network for object recognition” , submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. The dissertation author was a primary researcher and an author of the cited material.

The text of Chapter V, in full, is based on the material as it appears in: Sunhyoung Han and Nuno Vasconcelos, “Amorphous object detection in the wild” , in , 2011. The dissertation author was a primary researcher and an author of the cited material.

The text of Chapter VI, in full, is based on the material as it appears in: Sunhyoung Han and Nuno Vasconcelos, “Object-based regions of interest for image
compression”’ , in Data Compression Conference (DCC), 2008 and Sunhyoung Han and Nuno Vasconcelos, “Automatic detection of Object-based ROI (Region of Interest) for image compression”’ , submitted to IEEE Transactions on Image Processing. The dissertation author was a primary researcher and an author of the cited material.
VITA

1994-1998 Bachelor of Science, Electrical Engineering, Yonsei University, Seoul, Korea

1999–2000 Master of Science Electrical Engineering (VLSI & CAD), Yonsei University, Seoul, Korea

2000–2002 Researcher, Hynix Electronics

2002–2004 Senior Researcher, LetsVision

2005–2011 Research Assistant Statistical and Visual Computing Laboratory Department of Electrical and Computer Engineering University of California at San Diego

2004–2011 Doctor of Philosophy Electrical and Computer Engineering, University of California at San Diego

PUBLICATIONS


Sunhyoung Han and Nuno Vasconcelos. Automatic detection of Object-based ROI (Region of Interest) for image compression. to be submitted, IEEE Transactions on Image Processing.


ABSTRACT OF THE DISSERTATION

Hierarchical Discriminant Saliency Network for Object Recognition

by

Sunhyoung Han

Doctor of Philosophy in Electrical Engineering
(Signal and Image Processing)
University of California, San Diego, 2011
Professor Nuno Vasconcelos, Chair

Human visual perception mechanism is known to be effective and fast for object recognition problems and has inspired recognition algorithms. In this thesis we propose Hierarchical Discriminant Saliency Network (HDSN) mimicking hierarchical architecture of the primary visual cortex (V1). HDSN has feedforward hierarchical architecture tuned to goal-driven (top-down) recognition problem. First, we show a discriminant formulation of top-down visual saliency, intrinsically connected to the recognition problem. The formulation is shown to be closely related to a number of classical principles for the organization of perceptual systems, including infomax, inference by detection of suspicious coincidences, classification with minimal uncertainty, and classification with minimum probability of error. The resulting top-down saliency performs effectively as a focus of attention mechanism for the selection of interest points according to their relevance for visual recognition. Experimental results show that state-of-the-art computer vision algorithms works better when top-down saliency is used as preprocessor by pruning interest points. Then, stand alone discriminant saliency network based on discriminant saliency principle is presented. The biological plausibility of building blocks in the network, statistical inference and learning, tuned to the statistics of natural images, is investigated. It is shown that a rich family of statistical decision rules,
confidence measures, and risk estimates, can be implemented with the computations attributed to the standard neurophysiological model of V1. In particular, different statistical quantities can be computed through simple re-arrangement of lateral divisive connections, non-linearities, and pooling. It is then shown that a number of proposals for the measurement of visual saliency can be implemented in a biologically plausible manner, through such rearrangements. This enables the implementation of biologically plausible feedforward object recognition networks that include explicit saliency models. The potential of combined attention and recognition is illustrated by replacing the first layer of the HMAX architecture with a saliency network. Various saliency measures are compared, to investigate whether 1) saliency can substantially benefit visual recognition, and 2) the benefits depend on the specific saliency mechanisms implemented. Experimental evaluation shows that saliency does indeed enhance recognition, but the gains are not independent of the saliency mechanisms. Best results are obtained with top-down mechanisms that equate saliency to classification confidence. Finally, a novel biologically plausible hierarchical saliency network for visual recognition is proposed. Both of the layers are an optimal top-down saliency module, for the detection of a visual class of interest. The relationships between the proposed saliency network and existing solutions are discussed, for both convolutional network models, and more generic computer vision methods. This leads to some interesting insights, such as a mapping of popular computer vision algorithms to network form into building blocks, which highlights important discrepancies on the evaluation of the two types of approaches and gives a way of evaluating various algorithms in its component level. An extensive experimental evaluation shows that the proposed saliency network outperforms all existing network models, and all computer vision models of comparable parameters for both object localization and classification tasks. We also demonstrate that discriminant saliency network is suitable for amorphous object detection where the object is specified with no defined shape or distinctive edge configurations and automatic detection of region-of-interest for
image compression with additional EM type saliency validation process.
Chapter I

Introduction
I.A Hierarchical Discriminant Saliency Network for Object Recognition

The effectiveness and speed of biological object recognition have long inspired computer vision algorithms. In neuroscience, a number of properties of cortical networks have been identified as important for recognition. In particular, biological networks consist of layers of simple and complex cells [75], whose computations follow a standard neurophysiological model [20]. Simple cells implement a combination of filtering, rectification, divisive contrast normalization, and sigmoidal non-linearity, which makes them selective to certain orientations. Complex cells pool information from multiple simple cells, producing a more invariant representation.

Cortical layers are also organized hierarchically. While the receptive fields of cells at the lower hierarchical levels resemble Gabor filters of limited spatial extent, cells at the higher layers have much more complex receptive fields, and pool information from larger regions of support [122, 125]. This makes them more selective and invariant than their low-level counterparts. Various recognition models are inspired on this architecture. Recent examples include the HMAX network [129, 138, 117] and the extended convolutional networks of [123, 80]. Both have greater biologically plausibility than classical sigmoidal networks [52, 89, 33], and better performance. Extensive experiments have shown that accounting for simple and complex cells [138], using normalization and rectification [80], or optimizing the sequence of these operations can lead to major recognition gains [124].

In computer vision, a number of other architectures have achieved good performance in standard object recognition benchmarks [92, 107, 174, 181, 60, 178, 48, 43, 12, 45, 32]. Although differing in terms of detail, these architectures have many commonalities. For example, they all start by extracting a bag of localized image descriptors, such as SIFT [92, 174, 181, 60, 178, 12] or HOG [45, 32], that measure orientation dominance at each image location. When compared to
classical features, such as wavelets [109], Gabor decompositions [33], of principal components [106, 179, 87], these descriptors have much stronger orientation selectivity. Their probability distribution is then represented by a probability density model, typically a vector quantizer composed of a set of representative visual words. Descriptors are then assigned to words, and pooled over multiple regions, to create a number of spatially localized descriptor histograms. This increases the invariance of the representation. Descriptor histograms are finally fed to a support vector machine (SVM), or some other classifier. Several variants of this SVM have emerged, with the development of kernels suited for many types of features [63, 93, 13] and methods for multi-kernel classification [92, 60].

While apparently disjoint, the findings from neuroscience and computer vision have a strong commonality. In neuroscience, the modern standard neurophysiological model strongly emphasizes the importance of normalization, cell non-linearities, and pooling. In computer vision, most of the recent gains follow from the normalization implicit in the orientation histograms at the core of modern descriptors [105, 32], and the new pooling strategies [174, 15]. In summary, it appears that a more realistic reproduction of the biology is the best explanation for the advances in object recognition. However, the details of this connection are poorly understood. Since the standard neurophysiological model does not come with an optimality principle, its components lack a well-defined functional justification. It is not clear what functionality the intermediate layers are serving, or why operations such as contrast normalization or pooling make a difference. It is also not clear how to optimize these intermediate layers, e.g. how to enable the discriminative training [9] which appears to be critical for the effectiveness of the biological solutions to recognition [61, 141, 121]. Finally, it is difficult to compare algorithms in network form, such as HMAX or convolutional networks, to algorithm in computer vision form, e.g. [174, 12].

In this work, we bridge this gap with recourse to the attention literature. In this literature, it is well known that two saliency mechanisms play a role in
recognition. The first is *stimulus driven*, based on the contrast of simple features such as color, orientation, or motion [78]. These features are extracted with low-level image processing operators, e.g. Gabor filters. The second is *task driven*, and based on more complex features, such as object fragments [155], proto-objects [168] or objects [43]. While classical computational models of saliency have focused more on the stimulus driven component [78, 98, 18, 18, 130, 86, 180, 58, 56], also known as *bottom-up saliency*, a number of principled models have more recently been proposed for the task-driven component, also known as *top-down saliency*. These tie the optimality of saliency judgements to object recognition performance. Examples include discriminant saliency [57], which equates saliency to optimal decision-making, Bayesian models for top-down visual search and recognition [39], and the guided search [172], biased competition [34] and optimal gains [118] models, where feature maps are globally biased to improve recognition performance.

There are several reasons to integrate attention and recognition algorithms. First, there is little dispute that the attentional system is involved in object recognition. Second, there are examples in the literature of *improved* performance by the inclusion of focus of attention mechanisms in a recognition architecture, e.g. the finding by [117] that even the addition of rough spatial selectivity to the units of an HMAX network can substantially improve recognition performance. Third, there is a *strong connection* between saliency and the standard neurophysiological model. In fact, under the assumption of a widely accepted model of natural image statistics, most statistical models of saliency, e.g. inverse probability [130, 180, 18], self or mutual information [101, 5, 6], or Bayesian inference [39, 7], have an *explicit mapping* to the neurophysiologic model [75, 21, 69, 20]. Hence, saliency algorithms are, in principle, *compatible* with the most successful computer vision architectures. Finally, the connection to saliency, provides the intermediate layers of a recognition architecture with a *functional justification* and an explicit *optimization criterion*, which can be used for *discriminant training*. This can lead to non-trivial improvements in recognition performance.
In this work, these goals are accomplished by the introduction of hierarchical discriminant saliency networks. These are hierarchical networks, where each layer implements a saliency detector based on features of increasing selectivity and invariance. Unlike previous saliency models, these networks can account for both stimulus-driven saliency, based on simple features, and task-dependent saliency, implemented with more complex features. Each saliency detector is derived from the discriminant saliency principle [56], giving each network layer a clear functional justification: the extraction, from the visual field, of the optimal information for discrimination between two hypothesis. When one of these is an object class of interest, and the other the set of natural images, the network layers are automatically optimal for object recognition. Discriminant saliency also provides a precise optimality criterion for this task, the minimization of recognition error, and an optimal solution, the top-down saliency measure of [57]. Each of the proposed network layers implements this measure with the biologically plausible computations of [56].

This leads to a precise sequence of operations, which is consistent with the standard neurophysiological model [20], and gives the model a computational justification, assigning a precise statistical meaning to all intermediate operations. All parameters can thus be learned optimally with standard statistical procedures, enabling the explicit optimization of the network for recognition. Finally, the connections to the neurophysiological model are exploited to map several algorithms from the computer vision literature into network form, enabling a fair comparison both among themselves, and with the biologically inspired approaches. This identifies which attributes of the recognition architecture have a significant impact on recognition performance, and unveils some novel insights. For example, it is shown that, in addition normalization and pooling, the non-linearities implemented by the network can substantially impact its performance. In particular, it is shown that the ability to account for feature absence is critically important for some recognition problems. Finally, it is shown that common beliefs about the relative
performance of different recognition architectures are inaccurate, mostly the result of unfair experimental comparisons, due to the lack of a precise protocol for joint control of the parameters of all architectures.

I.B Contributions of the thesis

Discriminant saliency based on top-down decision theoretic principles provides optimal computational model having biological plausibility exploiting natural image statistics. In this thesis, we provide complete discriminant saliency network mimicking known operation in V1 visual cortex. We first show top down discriminant saliency principle tuned to given object recognition task and the biological plausibility of computational model for each component. Finally we provide full hierarchical discriminant saliency network for recognition problem and show improved performance on a variety of object recognition problems and image compression application. The main contributions of the thesis are as follows.

I.B.1 Top-down discriminant saliency

We propose a generic principle for top-down saliency, denoted by discriminant saliency, which is intrinsically connected to the recognition problem. We start from the intuition that, for recognition, the salient features of a visual class are those that best distinguish it from all other visual classes of recognition interest. This intuition translates naturally into a computational principle for the design of top-down saliency detectors: classification with minimal expected probability of error. It is shown that this principle is closely related to a number of previously proposed principles for the organization of perceptual systems: maximization of information transmission across perceptual layers (infomax) [101, 5, 6], inference by detection of suspicious coincidences [7], and classification with minimal uncertainty. For visual saliency, these principles equate optimal features to those maximally informative of presence/absence of the target class in the field of view,
whose observation is most suspicious in the absence of the target class, or which minimize the uncertainty about that presence/absence.

We investigate how these principles can be implemented with computational parsimony, by exploiting known properties of natural image statistics. It is shown that Barlow’s principle of inference by detection of suspicious coincidences [7] enables computationally efficient saliency measures that are nearly optimal in the minimum probability of error sense. Barlow’s principle is then used to derive computationally efficient algorithms for the two fundamental operations of discriminant saliency: feature selection and saliency detection. Experimental evaluation on object recognition tasks shows that the resulting top-down saliency detector can effectively act as a focus-of-attention mechanism, capable of pruning away bottom-up salient locations which are irrelevant for recognition. It is shown that this pruning improves the performance of state-of-the-art object recognition systems, in terms of both localization and classification accuracy. Finally, we show that discriminant saliency can adapt to a rich set of visual attributes.

I.B.2 Biological Plausibility of top-down discriminant saliency

We suggest a modification of the HMAX architecture that makes the connection between recognition and visual saliency explicit. We start by investigating the biological plausibility of statistical inference and learning tuned to the statistics of natural images. Building on prior work by [56], we show that a rich family of statistical decision rules, confidence measures, and risk estimates, can be implemented with the computations attributed to the standard neurophysiological model of V1 [75, 21, 69, 20]: a combination of linear filtering, divisive normalization, non-linearities, and spatial pooling. In fact, it is shown that all these computations have precise statistical meaning, contributing to an overall probabilistic interpretation where simple cells compute posterior probabilities and complex cells estimate statistical risks. It follows that a number of statistical operators can be implemented with biological hardware, through simple re-arrangement
of lateral divisive connections, non-linearities, and pooling. We next establish a connection to saliency mechanisms, by showing that various proposals for the measurement of visual saliency, from both the biological and computer vision literatures, can be implemented with biologically plausible reconfigurations of the standard neurophysiological model. By replacing the first layer of the HMAX architecture with these saliency networks, we conduct a rigorous experimental study of three questions at the intersection of attention and feedforward object recognition: 1) whether saliency benefits visual recognition, 2) whether the gains depend on the type of saliency considered (e.g. top-down vs bottom-up) or even the specific saliency algorithms, and 3) whether max-based pooling has an advantage over the classical linear operator. We note that the goal is not to investigate whether saliency is beneficial as a means to serialize recognition when there are multiple objects within the field of view, as has been done in [111, 168], or whether there are are gains in complementing recognition with an independent saliency path. Instead, we consider the question of whether saliency is intrinsically important for recognition, even when there is a single object in the field of view, as is suggested by computer vision research. Or, in other words, whether in addition to its predominant role within the “where” pathway, saliency also plays a role within the “what” pathway of object recognition. It is shown that the addition of saliency can significantly improve recognition performance, but that this is not independent of the saliency principle adopted. Best results are obtained with top-down saliency mechanisms that equate saliency to classification confidence.

I.B.3 Discriminant saliency network

Complete discriminant saliency network which enables hierarchical attention mechanisms is accomplished by the introduction of stand-alone discriminant saliency network. This is hierarchical network, where each layer implements a saliency detector based on features of increasing selectivity and invariance. Unlike previous saliency models, these networks account for both preattentive saliency
based on simple features, and attentive saliency implemented with more complex features. Each saliency detector is derived from the discriminant saliency principle [56], giving each network layer a clear functional justification. Discriminant saliency also provides a precise optimality criterion for this task, the minimization of recognition error, and an optimal solution, the top-down saliency measure of [57]. Each of the proposed network layers implements this measure with the biologically plausible computations of [56]. This leads to a precise sequence of operations, which is consistent with the standard neurophysiological model [20], and gives the model a computational justification, assigning a precise statistical meaning to all intermediate operations. All parameters can thus be learned optimally with standard statistical procedures, enabling the explicit optimization of the network for recognition.

The performance of the proposed discriminant saliency network architecture is evaluated in three ways. The first is a comparison to other saliency models on an object localization task, where it is shown to achieve the best top-down saliency results reported to date. Along with standard dataset, we consider amorphous object localization where the absence of low level feature compared to the surrounding scenes is distinctive. Discriminant saliency network equating the absence of feature as discriminancy shows significant improvement compared to state-of-arts methods. The second is a comparison to the previous biologically inspired recognition models on an object recognition task. It is shown that the addition of a clear optimality criterion for network optimization, as well as the clear functional justification provided by saliency, lead to substantial improvements over the best results in the literature. The third is a comparison to the the state-of-the-art recognition algorithms in the broader computer vision literature. Here, it is shown that several popular approaches in the literature can actually be mapped to a network form that resembles the one now proposed. This enables direct comparison by setting of comparable parameters the same. The results suggest that computer vision models achieve gains simply relying on extensions of computing
units which is not necessarily related to model itself. With the comparable computing units discriminant saliency network architecture showed better performance than state-of-the-art algorithms.

I.B.4 Application: Amorphous object detection in the wild

In this work, we investigate the detection of a large class of objects not covered by existing database portfolio. These are objects that lack many of the features commonly used as cues for object detection, and which we denote amorphous. Strictly speaking, amorphous objects have no distinctive edge configurations, texture, or a well defined shape. They can be found in science fiction movies, in the form of jelly-like creatures that can take any desirable shape. While, in the real world, truly amorphous objects are rare, many real objects are close to amorphous (e.g. a jellyfish, a bean bag, etc.), and an even larger set quasi-amorphous. By this, we refer to objects that can have very characteristic appearance under some canonical poses, but appear amorphous under others. Many such examples exist in the animal world. While the Panda face is very iconic, faceless poses tend to be quite amorphous.

We applied discriminant saliency network to the amorphous object detection. The intuition is that (at least in the natural world) the most distinctive property of amorphous objects is their lack of low level features, when compared to the surrounding scenes. This suggests modeling these objects as blobs of feature absence, i.e. regions where features that are usually active for natural images have a much weaker object response. One possibility for amorphous object detection is thus to rely on discriminant blob detection, by identifying blob-like regions in the responses of a set of features that are discriminant for object detection. A potential problem is, however, that discrimination may be due to the absence of feature responses. We overcome this problem by formulating blob detection as a form of top-down discriminant saliency.

Under this approach, detection is based on a two level classification ar-
architecture which implements a combination of feature-based and template-based discriminant saliency. The first level consists of a feature-based top-down saliency model, tuned for the detection of the target object. It is a robust classifier, that can detect the absence of a set of features, if this absence is informative of object presence. However, it is not highly selective, frequently generating false positives in background image regions. The second level learns discriminant templates of saliency response, which are then used to detect blobs of saliency compatible with the target object. This is again implemented with a top-down discriminant saliency model, tuned for object detection, which operates on saliency templates rather than image features. Altogether, this classifier is selective, yet robust enough to detect highly deformable objects of reduced visual structure. The use of saliency, rather than appearance, templates also makes it robust to pose variation.

I.B.5 Application: Automatic region-of-interest detection for image compression

We pursue an alternative strategy for the design of ROI detectors. This strategy is of a top-down nature, but trades the emphasis on highly accurate classification (characteristic of classical detector design) for an emphasis on 1) weak supervision and 2) learning efficiency. It consists of two stages: saliency detection and saliency validation. In the first, given an object class of interest, we search discriminant complex features (or filters) for the subset that best discriminates between 1) the examples from that class and 2) a set of generic images, representative of the distribution of all natural imagery. This can be done very efficiently, with the discriminant saliency network which has biological plausibility of human visual cortex and exploits the intermediate complex features coming from hierarchical architecture. Intermediate complex feature is shown to have superior performance compared to simple or more complex features [155]. Filtering an image with this discriminant complex features produce a discriminant saliency map, which has high magnitude in the locations of the class of interest and low
Of course, the cost of computational efficiency is some loss of classification accuracy. When compared to the output of a classical (strongly supervised) detector, discriminant saliency maps are less accurate in two ways. The first is smaller localization accuracy, i.e. ROI boundaries that are not as crisp as those obtained with strong supervision. This is not a significant problem for the applications considered here, since the relative increase in ROI area tends to be small. The second is a higher false-positive rate, i.e. the classification as salient of regions that may not contain objects of interest. This is a more significant problem, and is addressed by the saliency validation stage. This stage is inspired by recent developments in computer vision, which have shown the benefits of representing objects as constelations of “parts” [48, 4, 45]. The basic idea is to equate locations of maximal saliency with object “parts”, and build a model for the spatial configurations of these parts in the class of interest. The salient configurations of the image under analysis are then rated according to how well they are explained by this model. This imposes a constraint of geometric consistency between ROIs and the training examples, and is a powerfull filter for the rejection of false-positives.

We develop a computational efficient validation step by approximating all saliency maps by Gaussian mixtures, and using fast hierarchical inference procedures for model-building. While our geometric constraints are quite simple (much simpler than those commonly used in vision [4, 48, 45]), their combination with discriminant saliency is rather effective.

Overall, because we search for both the features and configurations that are more common in the object class of interest than in the generic class of natural images, the proposed strategy is fairly robust to the presence of clutter in the training images. This makes it possible to learn without the requirement for manual segmentation or alignment of examples during the assembly of the training set. The cost of tailoring the architecture for a new class is therefore quite low, making it possible for users to define new ROI detectors. The training sets used in
all of our experiments were automatically downloaded from the web, without any manual processing other than rejecting search engine errors, i.e. images that did not match the specified query. Furthermore, the architecture is completely generic, and applicable to any object category. We illustrate these properties by designing ROI detectors for image compression, in the context of various application scenarios that may be of real interest.

I.C Organization of the thesis

The rest of the thesis is organized as follows. In Chapter II, we present a generic top-down discriminant saliency which is intrinsically connected to the recognition problem. This chapter also provides usefulness of top-down saliency over bottom-up method by showing the attentiveness of saliency points as a preprocessor for existing computer vision algorithms. In Chapter III we investigate the biological plausibility of statistical inference and learning tuned to the statistics of natural images. We show that a rich family of statistical decision rules, confidence measures, and risk estimates, can be implemented with the computations attributed to the standard neurophysiological model of V1 [75, 21, 69, 20]: a combination of linear filtering, divisive normalization, non-linearities, and spatial pooling. Various proposals of for the measurement of saliency is rigorously investigated through experiments. In Chapter IV stand-alone discriminant saliency network is introduced and compared to existing computer vision algorithms. The performance of the proposed model is evaluated with various vision tasks, such as object localization, object classification, scene classification. In Chapter V amorphous object detection problem is introduced and discriminant saliency network is applied to the problem. Amorphous object detection problem has completely different characteristic to existing database portfolios in computer vision literature. In Chapter VI we present an application for automatic regions of interest detection for image compression. Finally, conclusions are provided in Chapter VII.
Chapter II

Top-down discriminant saliency
II.A  Saliency principles

We start by introducing the concept of discriminant saliency, and relating it to previous principles for the organization of perceptual systems.

II.A.1 Discriminant saliency principles

Discriminant saliency is rooted in a decision-theoretic interpretation of perception. Under this interpretation, perceptual systems evolve to produce decisions about the state of the surrounding environment that are optimal in a decision-theoretic sense, e.g. that have minimum probability of error. To achieve this goal, discriminant saliency is defined with respect to two classes of stimuli: a target class, and a null hypothesis, composed of all the stimuli that are not salient. The locations of the visual field that can be classified, with greatest confidence, as containing target stimuli are denoted as salient. This definition of saliency is applicable to a broad set of problems. For example, different specifications for target stimuli and null hypothesis enable its specialization to both bottom-up and top-down saliency. We have previously studied discriminant saliency in the context of bottom-up saliency detection, by combining it with center-surround image processing, and shown that the resulting detector is biologically plausible and replicates various results from the psychophysics of human saliency [56, 53]. In this work, we consider the implementation of top-down discriminant saliency, and its benefits for recognition.

For this, we define top-down saliency detection with respect to a one-vs-all classification problem, where target stimuli are drawn from an object class of interest, and the null hypothesis is composed of the stimuli drawn from all other object classes that make up the recognition problem. Visual stimuli are not measured directly, but through their projection onto a set of basis functions, or visual features. These features can be seen as matched detectors to certain attributes of the visual stimulus. The features that best discriminate between
target and null hypotheses are deemed salient. These are matched detectors to the salient attributes of the target. Salient locations are then defined as the locations of the visual field where the classification of the visual stimulus into target and non-target can be made with highest confidence.

When compared to bottom-up saliency, this definition has three interesting properties. First, by definition of target and null hypotheses, it makes saliency contingent upon the recognition problem, tuning it to the image attributes that best distinguish target from other object classes. Second, for a given object class, the saliency of a set of visual attributes changes with the recognition context. As the null hypothesis varies, so do the attributes that determine saliency. This is consistent with biological perception. For example, as illustrated by II.A.1, when a white fox is viewed against a forest, its color is salient and recognition is easy. On the other hand, when the fox is presented against white snow, color is no longer a salient attribute and recognition becomes very difficult. Third, and perhaps most importantly, discriminant saliency easily translates into an optimality criterion for the design of saliency algorithms. This design consists of two steps. The first is an optimal feature selection problem: the identification of the visual features that optimally discriminate between target and null hypothesis. The second is an optimal decision-making problem: the identification of the locations, in the visual field, where the presence of these features can be most confidently attributed to the target class. Both feature discrimination and classification confidence can be quantified in various ways, most of which are directly connected to previously proposed principles for the organization of perceptual systems.

II.A.2 The minimum Bayes error principle

For a classification problem defined by a feature space $\mathcal{X}$, and a random variable $Y$ that assigns $x \in \mathcal{X}$ to one of $M$ classes, $i \in \{1, \ldots, M\}$, the minimum probability of classification error is achieved by the Bayes classifier [38]

$$g^*(x) = \arg \max_i P_Y(i|x). \quad (\text{II.1})$$
Figure II.1 Feature saliency depends on the viewing context. White color is salient feature in the second image but it is not in the salient in the first image.

This probability of error is denoted as the Bayes error (BE)

\[ L^* = 1 - E_x[\max_i P_{Y|X}(i|x)], \quad (II.2) \]

where \( E_x \) is the expectation with respect to \( P_X(x) \). Since 1) the BE depends only on \( X \), 2) lower bounds the probability of error of any classifier on \( X \), and 3) there is at least one classifier (the Bayes classifier) that achieves this lower bound, minimization of BE is a natural optimality criteria for feature selection. Its minimization is, however, difficult due to the non-linearity of the \( \max(\cdot) \) operator in (II.2).

To relate the minimization of BE to other discriminant principles, we note that, \( P_{Y|X}(i|x) \) can be interpreted as a measure of confidence with which \( x \) can be assigned to class \( i \). Defining \( c(x) \) as the confidence measure for the classification of \( x \), it follows that, under Bayes decision theory,

\[ c^*(x) = \max_i P_{Y|X}(i|x). \quad (II.3) \]

The (optimal) decision rule is then to select the class of highest confidence, and optimal feature selection corresponds to the choice of \( X \) which maximizes the expected confidence on the classification decisions, \( E_X[c(x)] \). Under Bayes decision theory this expected confidence is

\[ C^* = E_X[c^*(x)] = E_X[\max_i P_{Y|X}(i|x)], \quad (II.4) \]

and its maximization is equivalent to the minimization of the Bayes error.
II.A.3 Relaxation and the principle of minimum uncertainty

The analysis above suggests a procedure to obtain consistent decision rules and feature selection costs: define a confidence measure $c(x)$, a feature selection cost $E_X[c(x)]$, and a decision rule which selects the class of highest confidence. One possibility to eliminate the non-linear $\max(\cdot)$ operator inherent to the BE, is to relax it, replacing $\max(\log p, \log(1-p))$ by $\text{mean}(\log p, \log(1-p)) = p \log p + (1-p) \log(1-p)$. We refer to this procedure as a relaxation to the mean.

Noting that Bayes decision rule is identical to

$$g'(x) = \arg \max_i \log P_Y(i|x), \quad (II.5)$$

the application of relaxation (to the mean) to this rule leads to the confidence measure

$$c'(x) = -H(Y|X = x), \quad (II.6)$$

where $H(Y|X = x) = -\sum_i P_{Y|X}(i|x) \log P_{Y|X}(i|x)$ is the entropy of the class label $Y$ given the observation $X = x$. We say that $c'(x)$ is consistent with Bayes decision rule, up to a relaxation to the mean. In this case, the expected confidence $E_X[H(Y|X = x)] = -H(Y|X)$ is the negative of the posterior entropy of $Y$ given $X$. Under this criterion, the optimal $X$ is the one which minimizes the uncertainty of the classification decision, where uncertainty is measured in the standard information theoretic sense (entropy). It follows that uncertainty minimization is equivalent to the minimization of Bayes error, up to a relaxation to the mean.

II.A.4 Infomax and Barlow’s principle of suspicious coincidences

Using the well known property that

$$I(Y; X) = H(Y) - H(Y|X), \quad (II.7)$$

where

$$I(X; Y) = \sum_i \int p_{X,Y}(x, i) \log \frac{p_{X,Y}(x, i)}{p_X(x)p_Y(i)} dx \quad (II.8)$$
is the mutual information between class label \( Y \) and feature vector \( X \) and \( H(Y) = -\sum_i P_Y(i) \log P_Y(i) \) the class entropy, and the fact that \( H(Y) \) does not depend on \( X \), it follows that uncertainty minimization is equivalent to selecting the features that have largest mutual information with the class label [8, 160, 165, 158]. This is frequently referred as the \textit{infomax} criteria, due to its connections to the \textit{infomax} principle for the organization of perceptual systems, a principle of long traditions in cognitive science [101, 5, 6]. The underlying confidence measure

\[
c''(x) = I(Y; X = x) = \sum_i P_{Y|X}(i|x) \log \frac{P_{X,Y}(x,i)}{P_X(x)P_Y(i)}, \tag{II.9}
\]

is a relaxation (to the mean) of the decision rule

\[
g''(x) = \arg \max_i \log \frac{P_{Y|X}(i,x)}{P_Y(i)P_X(x)}. \tag{II.10}
\]

This decision rule was proposed, by Barlow, as the fundamental computation of cerebral cortex [7]. He argued that a nerve cell “represents a hypothesis about the sense organs it connects with” and “the multitude of nerve cells in sensory pathways constantly test a multitude of hypotheses about the environment.” He then equated the cortex to a detective that “makes inductive inferences about the environment” by “looking out for suspicious coincidences. The occurrence of two events \( A \) and \( B \) is suspicious if they occur jointly more often than what would be expected from the probabilities of individual occurrence, i.e. \( P(A,B) \gg P(A)P(B) \).” “The cortical neurons in a region share amongst themselves the task of detecting the coincidences that occur on the input fibers of that region.” In computer vision, the detection of suspicious, or non-accidental, coincidences has been proposed as a principle for perceptual organization by various authors [11, 104].

It follows from the equivalence between infomax and uncertainty minimization that infomax is equivalent to the minimization of Bayes error, up to a relaxation to the mean. The different combinations of decision rule, confidence measure, and feature selection cost are summarized in Table II.1. It is quite interesting that, although the decision rules of Bayes and Barlow are different (one
Table II.1 Decision rules, confidence measures, and feature selection costs with different principles.

<table>
<thead>
<tr>
<th>rule</th>
<th>confidence</th>
<th>cost</th>
<th>principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayes</td>
<td>$g^*(x)$</td>
<td>$c^*(x)$</td>
<td>$-L^*$</td>
</tr>
<tr>
<td>Bayes (relaxed)</td>
<td>$g'(x)$</td>
<td>$-H(Y</td>
<td>X=x)$</td>
</tr>
<tr>
<td>Barlow (relaxed)</td>
<td>$g''(x)$</td>
<td>$I(Y;X=x)$</td>
<td>$I(Y;X)$</td>
</tr>
<tr>
<td>parsimony</td>
<td>$g''(x)$</td>
<td>$\sum_k I(Y;X_k=x)$</td>
<td>$\sum_k I(Y;X_k)$</td>
</tr>
</tbody>
</table>

minimizes error probability, the other seeks maximally suspicious coincidences), their relaxation produces identical feature selection costs (up to the constant $H(Y)$ which does not depend on $x$). In this sense, they are identical for feature selection.

II.B Learning salient features

Since the goal of top-down saliency is to identify regions for further processing, it should be computationally efficient. This implies that the selection of optimally discriminant features should itself be subject to constraints of computational parsimony. We investigate how to enforce such constraints by exploiting statistical properties of natural image features.

II.B.1 Image statistics and computational efficiency

One appealing property of infomax feature selection, and a substantial advantage over minimum BE, is its potential for computational parsimony. Defining $X_{1,k} = (X_1, \ldots, X_k)$, (II.8) can be rewritten as

$$I(Y;X) = \sum_k I(Y;X_k) + \sum_k [I(X_k;X_{1,k-1}|Y) - I(X_k;X_{1,k-1})]$$

where

$$I(X;Y|Z) = \sum_i \int P_{X,Y,Z}(x, i, z) \log \frac{P_{X,Y|Z}(x, i|z)}{P_X(x)p_Y(i|Z)} dx dz.$$  

The terms $I(X_k;Y)$ quantify the discriminant information conveyed by each feature, and the terms $I(X_k;X_{1,k-1}|Y) - I(X_k;X_{1,k-1})$ quantify the discriminant information contained in feature dependencies [158]. This can be combined with
Attneave’s hypothesis [5] that perception is tuned to the environment, to achieve substantial reductions in complexity. Of particular interest is a known statistical property of band-pass features, such as wavelet coefficients, extracted from natural images: that such features exhibit strongly consistent patterns of dependency across a very wide range of natural image classes [19, 73]. For example, when a natural image is subject to a wavelet decomposition, the conditional distribution of any wavelet coefficient, given the state of the co-located coefficient of immediately coarser scale (known as its “parent”), invariably has a bow-tie shape [19]. It follows that, while the coefficients are statistically dependent, their dependencies carry little information about the image class [19, 158]. Hence, the second summation of (II.11) is much smaller than the first, and (II.8) is well approximated by

$$I(X;Y) \approx \sum_k I(X_k;Y).$$

(II.13)

Note that this approximation does not assume that the features are independently distributed, but simply that their dependencies are not informative about the class. This is a new feature selection cost, and the expectation of a new confidence measure

$$c''(x) = \sum_k I(Y;X_k = x_k),$$

(II.14)

which results from relaxing (to the mean) the decision rules

$$g_k''(x) = \arg \max_i \log \frac{P_{Y,X_k}(i,x)}{P_{X_k}(x)P_Y(i)}, k \in \{1, \ldots, K\}.$$  

(II.15)

Note that this is a set of decision rules which act on the feature channels individually. We refer to this type of independent application of a decision rule to each channel as a marginal decision rule. The adoption of (II.13) enables two substantial simplifications. First, because mutual information is always positive, a very simple feature selection strategy is globally optimal when (II.13) holds [160]: to select the $K$ optimal features it suffices to 1) order all features by decreasing $I(X_k, Y)$, and 2) select the first $K$. Second, the terms in the right hand side of (II.13) only
require marginal density estimates. As we will see in the next section, these are extremely simple for bandpass features extracted from natural images.

II.B.2 Computational parsimony and suspicious coincidences

Returning to Table II.1, there are two important points to note. The first is that there is no equivalent to (II.11) for the feature selection cost of (II.4). Due to this, although it would be possible to define a computationally parsimonious feature selection cost of the form

\[ C = \sum_k E_{X_k} \left[ \max_i P_{Y|X_k}(i|x_k) \right] \]

it does not necessarily follow that such a cost would be a good approximation to (II.4). In fact, the relaxation of the max appears to be a necessary condition for the conjunction of near optimality and computational parsimony. Whether this relaxation has to be to the mean is, at this point, not known. Second, while the relaxation of (II.15) is sufficient for parsimony, the latter does not necessarily hold for relaxations of all marginal decision rules. In particular, even though the maximization of \(-H(Y|X)\) and \(I(Y;X)\) both lead to the infomax solution for feature selection, (II.13) does not imply that \(H(Y|X) \approx \sum_k H(Y|X_k)\). Instead, it can be shown, by application of (II.7), that it is identical to \(H(Y|X) \approx \sum_k H(Y|X_k) - (K - 1)H(Y)\). The confidence measure associated with this approximation is still (II.14) and not the relaxation of the marginal Bayes decision rules

\[ g^*_k(x) = \arg \max_i \log P_{Y|X_k}(i|x), k \in \{1, \ldots, K\}. \] (II.16)

In this way, the constraint of computational parsimony breaks the connection between infomax feature selection and relaxations of Bayes decision rule. In fact, among all decision rules considered so far, only the marginal rules \(g''_k(x)\) of (II.15) are consistent (up to a relaxation to the mean) with (II.13). For this reason, we believe that the detection of suspicious coincidences is preferable to the explicit
minimization of error probability when there are constraints of computational parsimony, and adopt this principle in the remainder of the work.

Note that, at this point, we do not have a decision rule for the observed feature vector $\mathbf{x}$, but the collection of marginal decisions of (II.15). This is intuitive: in the absence of discriminant feature dependencies, the detection of globally suspicious coincidences simplifies into the detection of suspicious coincidences in each feature channel. It is also consistent with the psychophysics of human saliency, where it is well known that humans can easily detect differences between targets and distractors along a single dimension (e.g. different color), but not when they require a conjunction of features (e.g. differences in the conjunction of color and orientation) [152]. Finally, the absence of a holistic decision rule for $\mathbf{x}$ is not problematic for saliency. In general, search tasks are better served by a graded measure of confidence $c(\mathbf{x})$ on how salient each feature vector $\mathbf{x}$ is, than by a hard binary classification. When considering multiple salient locations, attention should first be deployed to the location that can be declared salient with greatest confidence. If the object is not found there, the location of the next largest confidence should be inspected, and so forth. This is the rationale behind all existing saliency detectors, which search through local maxima of some saliency function [78, 172]. It is also consistent with the mechanisms of inhibition of return commonly found in biological vision [126, 89]. An holistic confidence measure is well defined for the principle of suspicious coincidences, under parsimony constraints: it consists of the sum of marginal confidence measures of (II.14).

**II.B.3 The generalized Gaussian distribution**

Besides (II.13), a well known property of the statistics of natural images can be exploited to increase computational efficiency: that the probability densities of band-pass image features extracted from such images are well approximated by
a generalized Gaussian distribution (GGD) [114, 109, 73],

\[ P_X(x; \alpha, \beta) = \frac{\beta}{2\alpha \Gamma(1/\beta)} \exp \left\{ - \left( \frac{|x|}{\alpha} \right)^\beta \right\}, \quad \text{(II.17)} \]

where \( \Gamma(z) = \int_0^\infty e^{-t}t^{z-1}dt, t > 0 \), is the Gamma function, \( \alpha \) is a scale parameter, and \( \beta \) is a shape parameter. The parameter \( \beta \) controls the rate of decay from the peak value, and defines a sub-family of the GGD (e.g. Laplacian when \( \beta = 1 \) or Gaussian when \( \beta = 2 \)). The GGD has various interesting properties. First, various low-complexity methods exist for the estimation of the parameters \((\alpha, \beta)\), including the method of moments [140], maximum likelihood (ML) [35] and minimum mean-square-error [73]. We adopt the method of moments in what follows. The two parameters are estimated from

\[
\sigma^2 = \frac{\alpha^2 \Gamma(3/\beta)}{\Gamma(3/\beta)} \quad \text{and} \quad \kappa = \frac{\Gamma(1/\beta) \Gamma(5/\beta)}{\Gamma^2(3/\beta)}, \quad \text{(II.18)}
\]

where \( \sigma^2 \) and \( \kappa \) are, respectively, the variance and kurtosis of \( X \)

\[
\sigma^2 = E_X[(X - E_X[X])^2], \quad \text{and} \quad \kappa = \frac{E_X[(X - E_X[X])^4]}{\sigma^4}.
\]

This method has been shown to produce good fits to natural images [73].

Second, it leads to closed form solutions for various information theoretic quantities. For example, when both \( P_X|Y(x|i) \) and \( P_X(x) \) are well approximated by GGDs, the mutual information \( I(X; Y) \) has a closed form. This follows from

\[
I(X; Y) = \sum_i P_Y(i) KL[P_X|Y(x|i)||P_X(x)] \quad \text{(II.19)}
\]

where \( KL[p||q] = \int p(x) \log \frac{p(x)}{q(x)} dx \) is the Kullback-Leibler (KL) divergence between \( p(x) \) and \( q(x) \), and [35],

\[
KL[P_X(x; \alpha_1, \beta_1)||P_X(x; \alpha_2, \beta_2)] = \log \left( \frac{\beta_1 \alpha_2 \Gamma(1/\beta_2)}{\beta_2 \alpha_1 \Gamma(1/\beta_1)} \right) + \left( \frac{\alpha_1}{\alpha_2} \right)^{\beta_2} \frac{\Gamma((\beta_2 + 1)/\beta_1)}{\Gamma(1/\beta_1)} - \frac{1}{\beta_1}. \quad \text{(II.20)}
\]

It can also be shown that

\[
H(X|Y = i) = \frac{1}{\beta_i} + \log \frac{2\alpha_i \Gamma(1/\beta_i)}{\beta_i} \quad \text{(II.22)}
\]
and

$$I(Y; X = x) = s[g(x)] \log \frac{s[g(x)]}{\pi_1} + s[-g(x)] \log \frac{s[-g(x)]}{\pi_0}, \quad (\text{II.23})$$

where $s(x) = (1 + e^{-x})^{-1}$ is a sigmoid function, $\pi_i = P_Y(i)$ is the prior for class $i$, and $g(x) = \left(\frac{|x|}{\alpha_0}\right)^{\beta_0} - \left(\frac{|x|}{\alpha_1}\right)^{\beta_1} + T$, with $T = \log \frac{\alpha_0 \beta_0 \pi_1 \Gamma(1/\beta_0)}{\alpha_1 \beta_0 \pi_0 \Gamma(1/\beta_1)}$. We will make use of these closed forms to derive an efficient implementation of discriminant saliency.

**II.C Discriminant saliency**

The design of a discriminant saliency detector has two components: feature selection and saliency detection.

**II.C.1 Feature selection**

We have seen in Section II.B.1 that, given a space $\mathcal{X}$ of band-pass features extracted from natural images, the best $K$-feature subset can be selected by computing the marginal mutual informations $M_k = I(Y; X_k)$, for all $k$, and selecting the $K$ features of largest $M_k$. The marginal mutual informations can be computed efficiently with (II.19) and (II.21). One final issue is that none of the feature selection costs considered so far is asymmetric: in general, discrimination does not differentiate between situations where 1) the feature is present (strong responses) in the object class of interest, but absent (weak response) in the null hypothesis, and 2) vice versa. Although both cases lead to low probability of error, feature absence is less interesting for saliency, which is an inherently asymmetric problem.

However, detecting if a feature is discriminant due to presence or absence in the class of interest is usually not difficult. For generalized Gaussian features, it suffices to note that feature absence produces a narrow GGD, close to a delta function, while feature presence increases the variance of the distribution (see Figure II.2 for an example). Since a narrow GGD has lower entropy than one of larger variance, discriminant features which are absent from the class of interest fail the
II.C.2 Feature learning

It has long been known that the careful selection of visual measurements is important for the solution of most computer vision problems. In the area of visual recognition, there has been a recent emphasis on localized representations, i.e. measurements that have a relatively small region of image support. This simplifies the design of the subsequent recognizer stages, by constraining the dimensionality of the feature space in which they operate, and improves the robustness of the representation to geometric transformations, due to camera motion, pose variability, etc. There are two main types of localized representations, which we refer to as
features [161] and parts [116, 14, 31, 48], and both have been widely used in the recent recognition literature.

Part-based representations based on prototypical patches usually coming from key-point detector, that depict image-like structures have recently become quite popular and successful in the context of the representation of objects as constellations of parts [4] and whole image classification with “visual texture” [116, 31, 48]. These methods usually need huge amount of computation in dealing with various key points and clustering to find appropriate “bag-of-words”. Also, it cannot be guaranteed to find optimal words with limited number of training examples.

Since all orthonormal feature sets span the space of image neighborhoods, feature-based representations usually do not need learning for features, which are just selected from a small number of orthonormal families (wavelet [167], Gabor [33, 109], or localized Fourier type of decompositions [161]) Among the pre-defined number of features discriminant feature can be selected in infomax sense and the saliency detector using this discriminant features showed promising results being trained with cluttered scenes in minimal computation [59].

In this paper, based on the saliency framework with simple salient features, the algorithm to find better discriminant feature set by rotating any possible basis axis to the direction of maximum classification. By rotating feature basis axis, discriminancy is improved yet holding feature dimension. Support Vector Machine is used as classifier upon the histogram of the responses from the selected complex feature which is the rotated version of original basis. Classification results are reported with four objects PASCAL data set [41].

In the space spanned by \( n \) basis function, projection to one dimension would be more discriminant if the projection is tilted to maximize the discriminancy. The idea is that all of the \( n \) dimensional basis is recomputed so that maximizing the marginal diversity. As shown in Figure II.3 the projection to the rotated axis is better for the classification because it is easy to separate the two
classes in that direction.

Figure II.3  Distribution example showing maximum discriminancy with (a) original basis and (b) projection to rotated basis

Now, $Z$ is a new random process which is a combination of

$$X = (X_1, \ldots, X_n)^T$$

(i.e. $Z = \Phi^T X$, where $\Phi$ is $1 \times N$ vector with $||\Phi||^2 = 1$). The best projection direction having the most discriminant power is the combination of the original basis satisfying

$$\Phi^* = \arg \max_{\Phi} I(\Phi^T X; Y)$$  \hspace{1cm} (II.27)

However, computing $P_{Z|Y}(x|i)$ for all of the possible $\Phi$ is not feasible. For example, in the case of using 64 DCT coefficient as a base feature and quantizing the rotation with three levels, $3^{64} = 3.4 \times 10^{30}$ times of feature response calculations are needed because feature responses are not a linear function for different $\Phi$s. Instead of finding $\Phi$ in one step, $\Phi$ is approximated and refined with the following steps where only two features are considered at one time but does not lose dimensionality at each iteration:

1) Initially, Set $\Phi = \{\Phi_i|i = 1, \ldots, N\}$, $\Phi_i$ is $1 \times N$ matrix with 0 except for the $i^{th}$ component with 1

2) Find $\theta^*$ and $j^*$ satisfying

$$\{\theta^*, j^*\} = \arg \max_{\theta, j} I(\cos \theta \Phi_i^* X + \sin \theta \Phi_j X; Y)$$  \hspace{1cm} (II.28)
where $i^*$ is the feature which has maximum among single $\Phi$

$$i^* = \arg \max_k I(\Phi_k X; Y)$$  
$$= <KL[P_{\Phi_k X|Y}(x|y)||P_{\Phi_k X}(x)]>_Y$$  

3) After finding $\{\theta^*, i^*, j^*\}$ the $i^{th}$ and the $j^{th}$ features are replaced with the rotated one with $\theta^*$ i.e.

$$\Phi_i = \cos \theta^* \Phi_i + \sin \theta^* \Phi_j$$  
$$\Phi_j = -\sin \theta^* \Phi_i + \cos \theta^* \Phi_j$$

This process is repeated to the next possible feature combination until no other rotation has bigger mutual information. Finally, the feature is sorted with mutual information in descending order. Since each replacement does not offend to the orthogonality and keeps the dimension equally, one can have better basis after each of the iteration in terms of their discriminant power with the same number of basis.

**II.C.3 Saliency detection**

Under discriminant saliency, all saliency judgments are based on a measure of classification confidence $c(x)$. As before, the asymmetric nature of the saliency problem needs to be taken into account: feature vectors that can be very confidently classified as not belonging to the class of interest should not be declared salient. This is accomplished by introducing a decision rule which summarily eliminates such feature vectors. Both the confidence measure and the decision rule should be consistent with (II.13). The fact that, in Table II.1, only (II.14) and (II.15) satisfy this requirement, leads to the saliency measure

$$S_D(x) = \sum_{k=1}^{K} S_k(x_k),$$  

with

$$S_k(x) = \begin{cases} I(Y; X_k = x) & \text{if } x \in S_k \\ 0, & \text{otherwise,} \end{cases}$$  

(II.32)
This measure has various interesting properties. First, it implements Barlow’s principle of suspicious coincidences by 1) identifying features whose appearance in the field of view is suspiciously coincident with that of the object class of interest, (II.33), and 2) equating saliency to the associated (log) degree of suspicion

\[ S_k(x) = \langle \log \frac{P_{Y|X_k}(i,x)}{P_Y(i)P_{X_k}(x)} \rangle, \]

where \( \langle f(x) \rangle = \sum_i P_{Y|X}(i|x)f(x) \). The overall saliency measure \( S_D(x) \) is the cumulative degree of suspicion over all feature channels.

Second, it equates salient features to matched filters for the detection of the salient visual attributes of the object of interest. This follows from the facts that 1) \( S_k \) can be written as

\[ S_k = \left\{ x \left| P_{X_k|Y}(x|1) > P_{X_k|Y}(x|0) \right. \right\}, \quad (\text{II.34}) \]

and 2) GGD features which pass the test of (II.24) have a narrower \( P_{X_k|Y}(x_k|0) \) than \( P_{X_k|Y}(x_k|1) \) (see Figure II.2). In result, \( S_k \) is of the form

\[ S_k = \{ x \| x \| > t_k \}, \]

where \( t_k \) is a threshold that depends on the parameters of the two GGDs, and only regions of large magnitude feature response are considered salient. This implies that the features are matched to the visual stimuli considered salient.

Finally, it has an intuitive interpretation as a mechanism for the allocation of attention. This follows from rewriting \( S_k \) as

\[ S_k = \left\{ x \left| P_{Y|X_k}(1|x_k) > P_Y(1) \right. \right\}. \]

At first analysis, when compared with Bayes decision rule \( (P_{Y|X_k}(1|x_k) > 1/2) \), this appears sub-optimal for low-probability objects. It should, however, be noted that Bayes decision theory is broader than we have considered so far. While a threshold
of 1/2 minimizes the expected probability of error, this minimum is of interest only when false-positives (spurious detections) and false-negatives (undetected targets) have equal costs. When this is not the case, the threshold is determined by the ratio of the costs. In summary, there is no single “optimal” threshold: different thresholds are optimal under different cost structures. A threshold of $P_Y(1)$ ties the cost structure to the observer’s prior beliefs on target likelihood, making the observer more or less conservative according to these beliefs. Searches for very unlikely targets have a low threshold, and require the inspection of a large number of locations. On the other hand, many locations are summarily rejected in searches for very likely targets. If the number of inspected locations is interpreted as the amount of attention devoted to the scene, this is an intuitive search behavior: searches for rare targets require more attention than searches for frequent ones. It motivates the use of (II.31) as a focus-of-attention (FOA) mechanism.

II.C.4 Discriminant saliency as a focus-of-attention mechanism

In biology, visual saliency usually combines bottom-up and top-down saliency mechanisms (e.g. [172]). Top-down saliency can be seen as a FOA mechanism, which prunes away bottom-up salient points in regions unlikely to contain objects of interest. We have implemented this FOA strategy by combining discriminant saliency with classical bottom-up operators from computer vision. The details of the resulting top-down interest point detector are given in Algorithm 1. This return a list of interest points $l_m$ ordered by decreasing discriminant saliency values $S_D(x_m)$ as an output. Locations of bottom-up saliency are first identified with classic interest point detectors. Top-down saliency, with respect to the object class of interest, is then computed with (II.31). The two saliency channels are combined by weighing interest points according to the value of discriminant saliency at their center location. The weighted points are finally sorted by decreasing top-down saliency, and used for either object localization or image classification, as discussed in the following section. While the combination with interest point de-
Algorithm 1 Top-down interest point detection

Training: Given a set of features $X_k, k \in \{1, \ldots, N\}$, a set of images $T_i$ from the target class, a set of images $T_0$ from the null hypothesis, and a target number of features $K$.

for $k = \{1, \ldots, N\}$ do

Estimate GGD parameters of $P_{X_k|Y}(x|i)$, from responses of $X_k$ to $T_i, i \in \{0, 1\}$, using (II.18).

Check whether $X_k$ passes the test of (II.25). If not, discard $X_k$ and move to feature $k + 1$.

Estimate GGD parameters of $P_{X_k}(x)$, from responses of $X_k$ to $T_0 \cup T_1$, using (II.18).

Compute $I(X_k, Y)$, using (II.19) and (II.21).

end for

Output: return the $K$ features of largest $I(X_k, Y)$.

Saliency detection: Given a test image $I$, a set of $K$ discriminant features $X_k$ for the target class, and the GGD parameters of $P_{X_k|Y}(x|i), i \in \{0, 1\}$, and $P_{X_k}(x)$.

Determine a set of interest point locations $l_1, \ldots, l_M$, using standard interest point operators.

for $k = \{1, \ldots, K\}$ do

for $m = \{1, \ldots, M\}$ do

Compute the response $x_m$ of $X_k$ at location $l_m$ of $I$, and $P_{X_k|Y}(x_m|i), i \in \{0, 1\}$, using (V.1).

Compute $S_k(x_m)$, using (II.32), (II.34), and (II.23).

end for

end for

for $m = \{1, \ldots, M\}$ do

Compute the saliency value at $l_m$ with (II.31).

end for
tection is not the only possible application of discriminant saliency, it enables an objective evaluation of the benefits of discriminant saliency for computer vision. In particular, because interest points are commonly used in state-of-the-art object recognition systems, it suffices to measure how the localization/classification accuracy of the latter varies with the addition of top-down attentional pruning.

II.D Discriminant saliency detection

Given the set of discriminant features for the object of interest, image locations containing the object should be detectable by monitoring the amplitude of feature responses. We next derive an optimal procedure to achieve this goal.

II.D.1 Discriminant saliency map

Under the discriminant saliency formulation, locations that can be assigned to the class of interest with lowest probability of error are optimally salient. From Bayes decision theory, these are the locations where the posterior probability of the class of interest is largest. To maintain consistency with feature selection, we consider the information theoretic alternative to this criteria. This is the maximization of the mutual information between class $Y$ and observed feature vector $x$. Considering the following relationship between mutual information $I(X; Y)$ and conditional entropy $H(Y|X)$ [29],

$$I(X; Y) = H(Y) - H(Y|X),$$  \hspace{1cm} (II.35)

where $H(Y) = -\sum_i P_Y(i) \log P_Y(i)$ is the entropy of the class label and

$$H(Y|X) = E_X \left[ -\sum_i P_{Y|X}(i|x) \log P_{Y|X}(i|x) \right],$$  \hspace{1cm} (II.36)

we note that maximizing $I(X; Y)$ is equivalently minimizing $H(Y|X)$. Thus, discriminant saliency detection selects locations where the class posterior entropy given the observation of $x$,

$$H(Y|X=x) = -\sum_i P_{Y|X}(i|x) \log P_{Y|X}(i|x),$$  \hspace{1cm} (II.37)
is minimal. These are the locations where the uncertainty of the class label is smallest, given their feature responses.

As before, we face two problems: the complexity of the probability estimates and the need to discard locations that can be classified with high confidence, but belong to the null hypothesis. Noting that $H(Y|X) = E_x[H(Y|X = x)]$, the two problems can be solved as before. Assuming that feature dependencies are not discriminant, the class posterior entropy can be approximated by

$$H(Y|X = x) \approx \sum_k H(Y|X_k = x_k).$$

Features from the null class can be detected by testing whether

$$P_{Y|X}(1|x) \log P_{Y|X}(1|x) < P_{Y|X}(0|x) \log P_{Y|X}(0|x)$$  \hspace{1cm} (II.38)

which is equivalent to testing whether

$$P_{Y|X}(1|x) < P_{Y|X}(0|x)$$  \hspace{1cm} (II.39)

This leads to the saliency measure

$$S_D(x) = \sum_{k=1}^K S_k(x_k),$$  \hspace{1cm} (II.40)

where

$$S_k(x_k) = \begin{cases} 
\log 2 - H(Y|X_k = x_k), & \text{if } P_{Y|X_k}(1|x_k) > P_{Y|X_k}(0|x_k); \\
0, & \text{otherwise};
\end{cases}$$  \hspace{1cm} (II.41)

The constant $\log 2$ simply assures that saliency is non-negative.

The result of computing $S_D(x)$ for all $x$ in a test image is referred to as the saliency map of that image. The implementation of the discriminant saliency detector is illustrated in Figure II.4. An input image is first filtered with the features selected in the training stage, generating a collection of feature maps. Each feature map is then subject to the discriminant saliency measure of (II.41) to produce a feature saliency map. These maps are finally pooled, according to (II.40), to obtain the global saliency map.

---

1This equivalence can be verified by plotting the functions $p \log p - (1 - p) \log(1 - p)$ and $p - (1 - p)$ in the range $p \in [0, 1]$ and noting that they are both positive if and only if $p > 1/2$. 
II.E Object recognition with discriminant saliency

In this section, we discuss some implementation details of the detector tested in this work. They do not impact saliency detection performance in a substantial manner.

II.E.1 Candidate features for discriminant saliency

In our experience, effective saliency detection can be attained with various bandpass feature sets. We have obtained very similar results with Gabor filters, Haar wavelets, and the discrete cosine transform (DCT) [55]. In the implementation discussed here, we have simply adopted a *multi-scale extension of the DCT*. Each image was decomposed into a five-level Gaussian pyramid, and $8 \times 8$ DCT features computed at each location and scale. The so-called DC coefficient (average of the image patch) was discarded at all scales, so as to guarantee lighting invariance. As shown in Figure II.6, many of the DCT basis functions can be interpreted as detectors of perceptually relevant image attributes, including edges, corners, t-junctions, and spots.
II.E.2 Determining the number of salient features

To determine the number $K$ of features to select, we start by noting that, if the output of a saliency detector is highly informative about the presence (or absence) of the class of interest in its input images, it should be possible to classify the images (as belonging to the class of interest or not) by classifying the associated saliency maps. This suggests the use of a saliency map classifier to determine the optimal number of features, using standard cross-validation procedures. We rely on a very simple saliency map classifier, based on a support vector machine (SVM) [156], which is applied to the histogram of saliency values derived from each image. Figure II.5 presents some examples of the classification accuracy as a function of the number of features, for the classification experiment of Section II.F.3. Note that the accuracy is fairly constant over a large range of parameters. Visual inspection of saliency maps obtained with different cardinalities has also shown no substantial differences from the one obtained with the optimal number of features.

II.F Experimental evaluation

The performance of the proposed discriminant saliency detector (DSD) was evaluated on a set of weakly supervised object recognition experiments.
Table II.2 Number of training and testing images in 10 object classes PASCAL dataset.

<table>
<thead>
<tr>
<th></th>
<th>bicycle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>cow</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Training</td>
<td>270</td>
<td>174</td>
<td>553</td>
<td>386</td>
<td>206</td>
</tr>
<tr>
<td># of Testing</td>
<td>268</td>
<td>180</td>
<td>544</td>
<td>388</td>
<td>197</td>
</tr>
<tr>
<td>dog</td>
<td>365</td>
<td>247</td>
<td>235</td>
<td>666</td>
<td>251</td>
</tr>
<tr>
<td>horse</td>
<td>254</td>
<td>234</td>
<td>675</td>
<td>238</td>
<td></td>
</tr>
<tr>
<td>motorbike</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>person</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sheep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

II.F.1 Experimental setup

Datasets

Weakly supervised object recognition addresses the design of object recognition systems from informally collected examples. In particular, training examples of the object of interest are presented against cluttered backgrounds. Two tasks are usually considered. The first is to determine whether a given image contains an instance of the object of interest (classification). The second is to locate the image area covered by the latter (localization). For both tasks we adopt the PASCAL2006 dataset [40], which contains 10 object categories: bicycle, bus, car, cat, cow, dog, horse, motorbike, person, and sheep. It is an interesting dataset because the images were collected with limited control over appearance and pose of objects and background, and many images contain instances of several classes. It also provides ground truth for each object, since all images are annotated with bounding boxes. Although the ground truth was not used for training, its availability allows objective measurements of localization accuracy. The images were divided into training and test sets, according to Table II.2. For each category, we defined a one-vs-all classification problem opposing the object class under consideration to the others.
Figure II.6 Some of the basis functions in the DCT feature set. Many of the DCT basis functions can be interpreted as detectors of perceptually relevant image attributes, including edges, corners, t-junctions, and spots.

**Bottom-up interest points**

Bottom-up interest points are extracted with three operators widely used in object recognition: Harris-Laplace (HarrLap), Hessian-Laplace (HesLap) [112], and difference of Gaussian (DoG) [105]. The binaries are available from [http://lear.inrialpes.fr/people/dorko/downloads.html](http://lear.inrialpes.fr/people/dorko/downloads.html).

**Candidate features for discriminant saliency**

The discriminant saliency detector does not have free parameters. The only component left to be specified is the initial pool of band-pass features. We have obtained very similar results with Gabor filters, Haar wavelets, and the discrete cosine transform (DCT) [55]. The implementation discussed here is based on a *multi-scale extension of the DCT*. Each image is decomposed into a five-level Gaussian pyramid, and $8 \times 8$ DCT features computed at each location and scale. The so-called DC coefficient (average of the image patch) is discarded at all scales, to guarantee lighting invariance. As shown in Figure II.6, many of the DCT basis functions can be interpreted as detectors of perceptually relevant image attributes, including edges, corners, t-junctions, and spots.

**Preliminary analysis of contextual influences**

Since, in weakly supervised recognition, all images are presented with backgrounds, it is important to verify whether a classifier performs well because it has learned to recognize the objects of interest, or because it has learned to classify
the background. It is well known that background scenes provide a substantial amount of contextual information, which can significantly simplify the recognition tasks [150, 119, 178]. While, in general, the ability to exploit context is an asset, it is not the goal of saliency. On the contrary, saliency aims to identify the objects themselves, so that the recognizer can learn clutter-free object models. These could obviously be complemented by context models.

In PASCAL2006, context is a strong cue for the recognition of some object classes. For example, cows and sheep always stand on grass, while cars and buses are surrounded by roads, buildings, or parking lots. To quantify the strength of these contextual cues, we performed a preliminary image classification experiment, comparing two detectors. The first used the bottom-up interest points located inside the ground truth object bounding box (referred to as BU-GT). The second used those located outside the box (referred to as BU-BG). The classifier was that proposed in [178], and will be described in detail in Section II.F.3. Table II.3 presents the classification accuracy obtained for all classes. It is clear that, in some cases, the contextual cues are quite strong. For example, cow classification using merely background points (no object information) is only 3.56% less accurate than that using perfectly segmented interest points. Other classes for which strong contextual influences were observed include car, sheep, and bus.

These contextual influences create two problems for saliency. First, a saliency detector can only learn to separate object from background if the background scenes are diverse [159]. If a cow always appears in a patch of grass, it is impossible to learn that cow and grass are different visual concepts. For discriminant saliency, this translates into the selection of features (e.g., grass descriptors) which are discriminant because they enable the detection of the “object-attached” background (grass), not the object itself. Note that this is not a problem for classification (assuming, of course, that the test images show cows in the same grass patch), but it is a problem for localization. In this sense, localization is a better measure of saliency detection performance than image classification. Sec-
Table II.3 Image classification accuracy (%) of Bag-of-Feature Spatial Pyramid (BFSP) classifiers for PASCAL2006.

<table>
<thead>
<tr>
<th></th>
<th>bicycle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>cow</th>
</tr>
</thead>
<tbody>
<tr>
<td>BU-GT</td>
<td>96.18</td>
<td>99.34</td>
<td>97.89</td>
<td>94.32</td>
<td>92.64</td>
</tr>
<tr>
<td>BU-BG</td>
<td>81.62</td>
<td>91.63</td>
<td>92.51</td>
<td>86.25</td>
<td>89.08</td>
</tr>
<tr>
<td>Diff. (%)</td>
<td>14.56</td>
<td>7.71</td>
<td>5.38</td>
<td>8.07</td>
<td>3.56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>dog</th>
<th>horse</th>
<th>motorbike</th>
<th>person</th>
<th>sheep</th>
</tr>
</thead>
<tbody>
<tr>
<td>BU-GT</td>
<td>85.78</td>
<td>92.59</td>
<td>98.56</td>
<td>87.42</td>
<td>95.39</td>
</tr>
<tr>
<td>BU-BG</td>
<td>76.20</td>
<td>81.99</td>
<td>82.72</td>
<td>73.20</td>
<td>89.13</td>
</tr>
<tr>
<td>Diff. (%)</td>
<td>9.58</td>
<td>10.60</td>
<td>15.84</td>
<td>14.22</td>
<td>6.26</td>
</tr>
</tbody>
</table>

ond, a feature can be discriminant simply because it is consistently absent from the “object-attached” background. This is not a problem for discriminant saliency, since such features are rejected by the test of (II.24), unless the effect is overwhelming. For example, the ubiquity of large smooth regions (grass) in the cow class makes any textured feature discriminant for its detection. Since the animal body is also textureless, none of these features is informative for the class of interest, and all features are discarded. In PASCAL2006, we observed this behaviour for the cow and sheep classes. The fact that discriminant saliency rejects all features is actually one of its advantages: it indicates that the underlying feature pool is not rich enough to account for these objects. In this case, it should contain shape features, which are important cues for cow or sheep detection. While we could have pursued an expanded feature set, we felt that this issue is complementary to the discussion of the paper, and eliminated the cow and sheep classes from further consideration.

II.F.2 Object localization

We next evaluate the localization accuracy of the combination of bottom-up interest point detection and top-down pruning, based on DSD. We refer to the locations deemed salient by the combination of the bottom-up and top-down mechanisms as \textit{DSD-BU points}. The localization accuracy of DSD-BU is compared
to those of three discriminant interest point pruning strategies previously proposed in the literature.

**DVW, LSVM, and pLSA detectors**

The *discriminative visual words* (DVW) detector [16, 26] extracts interest points from a collection of training images, and describes them by SIFT region descriptors [105]. It then estimates the distribution of the SIFT descriptors with a Gaussian mixture model (GMM) and standard clustering techniques. Each cluster center is referred to as a *visual word*. Discriminant visual words are found with an estimate of the posterior probability of object $i$ given visual word $w$

$$D(w) = P_{Y|W}(i|w) = \frac{\text{# of times that } w \text{ appeared in images from class } i}{\text{# of times that } w \text{ appeared in any image}}$$

referred to as the discriminability of word $w$. The discriminability of an interest point is quantified by the posterior probability of the object given the SIFT descriptor ($x$) at the point

$$P_{Y|X}(i|x) = \sum_w D(w) P_{W|X}(w|x), \quad \text{(II.43)}$$

$$P_{W|X}(w|x) = \frac{P_{X|W}(x|w)P_W(w)}{\sum_w P_{X|W}(x|w)P_w(w)},$$

where $P_w(w)$, and $P_{X|W}(x|w)$ are the GMM components. All reported results were obtained with 3000 visual words, learned with K-means clustering.

Like the DVW detector, the *linear support vector machine* (LSVM) selects discriminant visual words [82]. However, discriminability is measured with recourse to a linear SVM which classifies histograms of visual words. Following [113], the discriminability of a feature (or visual word) is equated to the absolute value of the weight given to that word by the SVM. To account for feature discrimination due to presence or absence in the object class, the discriminability of visual words which appear more often in the null hypothesis than in the target class is set to 0, akin to the test of (II.24). During detection, the saliency of an interest point is
measured by the discriminability of the corresponding visual word. Once again, all experimental results were obtained with 3000 visual words learned with K-means.

The probabilistic Latent Semantic Analysis (pLSA) detector is a topic discovery method developed in the text analysis literature, which has been successfully applied to object categorization [143, 47]. It models each image as a document of visual words. The set of images is equated to a text corpus, and pLSA learns object categories as mixtures of representative topics for this corpus. Mathematically, an image (or document \(d\)) is represented as a “bag” of visual words \((w)\) sampled from a hidden topic variable \((Z)\). The joint distribution decomposes into

\[
P_{W,D,Z}(w,d,z) = P_D(d)P_{Z|D}(z|d)P_{W|Z}(w|z),
\]

where \(P_{W|Z}(w|z)\) and \(P_{Z|D}(z|d)\) are multinomial distributions. Estimation of the pLSA model involves determining topic vectors representative of all documents, and the coefficients of each document, i.e. finding topic-specific word-distributions \(P_{W|Z}(w|z)\) and document-specific mixing proportions \(P_{Z|D}(z|d)\). This is done with the expectation-maximization (EM) algorithm [143, 72]. The pLSA model was originally proposed for unsupervised learning, but it was shown that, for cluttered data, the learned object topics are likely to include words representative of background clutter [143]. To avoid this, we train the model in the discriminant manner suggested in [143]. A set of “background” topics is first learned from negative training images, and frozen. The topics representative of each object class are then learned from the set of images of that class. One pLSA model is learned for each object category. For the detection of an object category, bottom-up interest points are pruned by the topic posterior probabilities, \(P(z|w,d)\), at these points. Similarly to DVW and LSVM, the visual word dictionary consisted of 3000 words learned with K-means. The number of object and background topics were determined by cross-validation. Best results were achieved with seven background and one object topic. This was the configuration adopted to produce the results reported.
Measure of localization accuracy

To measure localization accuracy of pruned salient points we rely on precision/recall (PR) curves. Given a threshold, the points of saliency above it are classified as “object” and the remainder as ”background”. Denoting the ground-truth bounding box by $B$, the following quantities are computed for each object/background assignment:

- True Positive (TP) - number of “object” points with center inside $B$;
- False Negative (FN) - number of “background” points with center inside $B$;
- False Positive (FP) - number of “object” points with center outside $B$.

The recall and precision rates are then defined as

\[
\text{Recall} = \frac{TP}{TP + FN} \quad \text{Precision} = \frac{TP}{TP + FP}. \tag{II.45}
\]

Localization accuracy is quantified as the average precision (AP) over the range of recalls.

Results and discussion

Figure II.7 presents the PR curves produced by various detectors, on PASCAL2006. A plot is presented for each object category, comparing the PR curves of both the four top-down saliency detectors (DSD-BU, DVW, LSVM, and pLSA) and the three bottom-up interest point operators (HarrLap, HesLap, and DoG). The average precision (AP) of each detector is presented in Table II.4.

Three conclusions are supported by these results. First, all top-down pruning strategies improve the localization accuracy of bottom-up interest points, especially at low recall rates, indicating that top-down pruning concentrates interest points on regions informative of object presence. This is confirmed by Figure II.8, where we present saliency maps produced by DSD. All scenes contain instances from two object classes, e.g. (a) “person” and “car”, (b) “person” and
Figure II.7 Precision-recall of various saliency detectors on 8 object classes in PASCAL2006.

“bus”, (c) “motorbike” and ”car”, and (d) “person” and “motorbike”. In each case, we draw a bounding box around the object of interest and show, on the right, the saliency map for the detection of that object. It is clear that DSD successfully switches between the two objects, highlighting the one of interest and suppressing all others.

Second, among the four top-down detectors, DSD-BU clearly achieves the best localization accuracy. Its median AP, across the eight object classes, is at least 15% higher than those of DVW, LSVM, and pLSA. This suggests that all stages of a top-down saliency detector must be discriminant. The main difference between DSD and the three previous approaches is that, while DSD starts by selecting dis-
Table II.4  Average precision-recall for various detectors. DSD-BU achieves the best performance among others.

<table>
<thead>
<tr>
<th>Detector</th>
<th>bicycle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>dog</th>
<th>horse</th>
<th>motorbike</th>
<th>person</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSD-BU</td>
<td>0.76</td>
<td>0.72</td>
<td>0.60</td>
<td>0.78</td>
<td>0.76</td>
<td>0.75</td>
<td>0.82</td>
<td>0.56</td>
</tr>
<tr>
<td>DVW</td>
<td>0.65</td>
<td>0.61</td>
<td>0.45</td>
<td>0.64</td>
<td>0.60</td>
<td>0.50</td>
<td>0.67</td>
<td>0.34</td>
</tr>
<tr>
<td>LSVM</td>
<td>0.59</td>
<td>0.58</td>
<td>0.37</td>
<td>0.62</td>
<td>0.56</td>
<td>0.49</td>
<td>0.62</td>
<td>0.35</td>
</tr>
<tr>
<td>pLSA</td>
<td>0.63</td>
<td>0.56</td>
<td>0.43</td>
<td>0.62</td>
<td>0.58</td>
<td>0.49</td>
<td>0.66</td>
<td>0.34</td>
</tr>
<tr>
<td>HarrLap</td>
<td>0.57</td>
<td>0.51</td>
<td>0.33</td>
<td>0.67</td>
<td>0.59</td>
<td>0.46</td>
<td>0.60</td>
<td>0.36</td>
</tr>
<tr>
<td>HesLap</td>
<td>0.63</td>
<td>0.54</td>
<td>0.41</td>
<td>0.65</td>
<td>0.58</td>
<td>0.49</td>
<td>0.64</td>
<td>0.38</td>
</tr>
<tr>
<td>DoG</td>
<td>0.57</td>
<td>0.52</td>
<td>0.37</td>
<td>0.60</td>
<td>0.54</td>
<td>0.51</td>
<td>0.61</td>
<td>0.33</td>
</tr>
</tbody>
</table>

criminant features, DVW, LSVM, and pLSA cluster interest points. Since neither interest points nor clustering are discriminant, much of the information of interest for recognition is eliminated before any discriminant learning ever takes place. While visual words can be seen as complex features, this non-discriminant nature limits their localization performance. In fact, the results of Figure II.7 show that they can perform substantially worse than a simple orthogonal band-pass feature decomposition. Discriminant feature selection from a poorly discriminant feature set is not sufficient to guarantee good classification performance.

Comparing DVW and pLSA, we note that DVW achieves a slightly better average performance. Again, this suggests an advantage for explicitly discriminant representations. Although pLSA is trained in a “discriminant” manner, it is intrinsically a generative model with limited ability to distinguish objects of interest from clutter. The observation that DVW also performs somewhat better than LSVM indicates an advantage of soft (the probabilistic representation used by DVM) over hard (the histogram of visual words used by LSVM) assignments of interest points to visual words. Figure II.9 and Figure II.10 present saliency detection examples (one per object class, salient points at 40% recall) for the four top-down detectors. Clearly, bottom-up points pruned with DSD are more likely to be co-located with the target than those pruned with the other methods.

Finally, we note that although strong contextual influences exist for some classes, e.g. car and bus, DSD does not appear to have any problems learning
features informative of the object of interest, rather than the background. This indicates that the consistency of appearance among the objects in these classes is higher than that of the background, making object features more discriminant than background features. This leads to their selection by discriminant saliency.

II.F.3 Image classification

Classification performance was evaluated with a classification architecture that has been shown to achieve state-of-the-art results on PASCAL2006 [178, 40].

Object classifier

The adopted architecture combines the “bag-of-features” image representation with the spatial pyramid matching technique (and is referred to as the $BFSP$
Figure II.9 Salient locations at 40% recall. Top to bottom: original images (objects marked by their bounding boxes), salient locations pruned by DSD, DVW, LSVM, and pLSA. Each circle represents the location and size of a salient point. White (black) indicates that the salient point falls inside (outside) the ground truth bounding box for the object.
Figure II.10  Salient locations at 40% recall. Top to bottom: original images (objects marked by their bounding boxes), salient locations pruned by DSD, DVW, LSVM, and pLSA. Each circle represents the location and size of a salient point. White (black) indicates that the salient point falls inside (outside) the ground truth bounding box for the object.
classifier). Both its implementation and the selection of all parameters followed closely [40]. The bag-of-features representation is based on a dictionary of visual words, which are cluster centers of SIFT descriptors extracted at interest points locations from all training images. These can be either the original interest points, or the result of their pruning by top-down saliency. In all cases, a K-means algorithm was adopted for clustering and the dictionary contains 3000 visual words. The spatial pyramid representation is obtained by 1) repeatedly subdividing the image into increasingly finer sub-regions, and 2) characterizing each sub-region by a histogram of the visual words found inside it [93]. This representation is combined with a $\chi^2$ kernel to train an SVM for each object class.

**Results and discussion**

For compatibility with the results presented in the PASCAL2006 competition [40], classification accuracy was measured by the *Area Under ROC Curve (AUC)*. Table II.5 presents the accuracy achieved for the different object classes. Three classification results are listed for each object category. They were obtained by varying the set of interest points fed to both K-means clustering and classifier. The first, labeled “BU”, is the ensemble of all points generated by the interest point operators (HarrLap, HesLap, and DoG). This was the set of points adopted in the original implementation [40]. The second is the set of DSD-BU points, produced by pruning these interest points with discriminant saliency. To emphasize the areas of the objects of interest, saliency maps were first thresholded, by setting to zero all locations of saliency smaller than 10% of the maximum (measured over the entire map). The third is the set of interest points that fall inside the ground truth bounding box (BU-GT). The three sets are derived from the same pool of interest points, but result from different pruning strategies: (1) no pruning, (2) discriminant and practical pruning, and (3) perfect but unrealistic pruning.

As expected, the performance of BU-GT is usually the best. The only exception is the dog class where DSD-BU actually outperforms BU-GT. In all
Table II.5  Image classification accuracy (%) of Bag-of-Features Spatial Pyramid (BFSP) classifiers for PASCAL2006.

<table>
<thead>
<tr>
<th></th>
<th>bicycle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>dog</th>
<th>horse</th>
<th>bike</th>
<th>person</th>
</tr>
</thead>
<tbody>
<tr>
<td>BU</td>
<td>94</td>
<td>98</td>
<td>97.2</td>
<td>92.6</td>
<td>85.6</td>
<td>90</td>
<td>96.3</td>
<td>82.2</td>
</tr>
<tr>
<td>BU-GT</td>
<td>96.2</td>
<td>99.3</td>
<td>97.9</td>
<td>94.3</td>
<td>85.8</td>
<td>92.6</td>
<td>98.6</td>
<td>87.4</td>
</tr>
<tr>
<td>Gain (%)</td>
<td>2.2</td>
<td>1.4</td>
<td>0.7</td>
<td>1.7</td>
<td>0.2</td>
<td>2.6</td>
<td>2.3</td>
<td>5.2</td>
</tr>
<tr>
<td>DSD-BU</td>
<td>94.2</td>
<td>98.1</td>
<td>97.3</td>
<td>92.8</td>
<td>86</td>
<td>91</td>
<td>97.1</td>
<td>82.4</td>
</tr>
<tr>
<td>Norm. Gain (%)</td>
<td>8.8</td>
<td>11.6</td>
<td>17.8</td>
<td>10.9</td>
<td>209</td>
<td>38.7</td>
<td>36.1</td>
<td>3.7</td>
</tr>
</tbody>
</table>

cases, BU performs the worst. To quantify the improvements of DSD-BU over BU, we normalized the gain from BU to DSD-BU by that from BU to BU-GT, i.e. $\frac{\text{accuracy}(\text{DSD-BU}) - \text{accuracy}(\text{BU})}{\text{accuracy}(\text{BU-GT}) - \text{accuracy}(\text{BU})}$. This measure reflects the fact that BU-GT is expected to achieve the best performance (since it is based on perfect object segmentation) and quantifies the percentage of the gap between BU and BU-GT which is recovered by DSD-BU. As can be seen from the last row of Table II.5, the median value of this percentage gain across classes is 14.7. For classes of dog, motorbike, and horse, DSD-BU recovers more than 36% of the gap between BU and BU-GT. These results show that the proposed discriminant saliency detector captures relevant information for object classification. Finally, it is worth pointing out that, for all object classes, DSD pruned away at least 30% of the BU points. This can be a source of significant computational savings for some applications.

II.G  Conclusion

We have proposed a novel formulation for top-down saliency, denoted as discriminant saliency, which is intrinsically grounded on the recognition problem. Under this formulation, salient features are defined as those whose responses best discriminate between visual stimuli drawn from a target class and those drawn from a null hypothesis, composed of all other classes of possible recognition interest. Saliency is then defined as the confidence with which locations in the visual field can be classified as containing stimuli drawn from the target class. While both
discrimination and classification confidence can be defined with respect to a number of previously proposed computational principles for perceptual organization, we have argued for the adoption of Barlow’s principle of inference by detection of suspicious coincidences. In particular, it has been shown that the combination of this principle with known statistical properties of natural images enables computationally parsimonious implementations of both feature selection and saliency detection. The resulting discriminant saliency detector is quite effective as a “focus-of-attention” mechanism, which can prune the interest points commonly used in computer vision according to their relevance for recognition. Experiments were designed to evaluate the benefits of this FOA mechanism in object localization and image classification tasks. In both cases, the addition of discriminant saliency was shown to improve the performance of current state-of-the-art methods.

II.H Acknowledgments

The text of Chapter II, in full, is based on the material as it appears in: Dashan Gao, Sunhyoung Han and Nuno Vasconcelos, “Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition”, published in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009 and Sunhyoung Han and Nuno Vasconcelos, “Complex discriminant features for object classification”, published in International Conference on Image Processing, 2008. The dissertation author was a primary researcher and an author of the cited material.
Chapter III

Biologically Plausible Saliency Mechanisms Improve Feedforward Object Recognition
III.A Introduction

The effectiveness and speed of biological solutions to the object recognition problem have long been a source of inspiration for recognition algorithms. The introduction of the backpropagation algorithm [132] established a framework for the automated design of recognition networks, and was highly successful for a number of problems. In particular, convolutional networks were shown to be highly competitive with the best non-biological classifiers for tasks such as hand-written character recognition [94]. More recent results, by Thorpe and collaborators [148], on the ability of human subjects to categorize natural scenes, showed that such tasks can be performed with high accuracy (close to 94%) and very quickly (in less than 150 ms). The fact that such low recognition times leave no room for propagation of feedback across cortical areas, reinforced the significance of feed-forward networks in visual recognition, at least in its early stages. It also spurred a renewed interest in the family of feedforward architectures, of which the most recent popular element is the HMAX network of [129, 138]. This network emulates the organization of the visual system as a cascade of layers of simple and complex cells [75], and has been recently shown to achieve state-of-the-art performance for a number of recognition tasks [117].

There are, however, two important limitations of the HMAX model. First, because the organization of the network lacks a clear computational justification, HMAX networks also lack a principled optimality criterion and training algorithm. This limits their relevance as an explanation for the underlying biological computations. Second, HMAX networks do not account for the psychophysical evidence on the important role played by visual attention in top-down processes such as object recognition [176]. This limitation has been somewhat mitigated by research on recognition within multi-object displays, which complements the HMAX network with serial attention mechanisms [111, 168]. In these methods, saliency is computed with an independent bottom-up network, which 1) acts as a “front-end”
to the HMAX network, selecting patches of the visual field to recognize [111], or 2) modulates the connections of some HMAX units, serially directing attention to different proto-objects in the field of view [168]. None of these works can account for the role of top-down attention in recognition, or the benefits of saliency in single object displays. These benefits have been documented in the computer vision literature [83, 137, 112], but with recourse to interest-point detectors that are not biologically plausible. Within the HMAX literature, it has been shown that limiting the spatial pooling performed by some of the HMAX units can lead to non-trivial recognition improvements [117]. This, however, has been done in a somewhat ad-hoc form, by restricting the receptive fields of these units to a pre-defined window size. To the best of our knowledge, no formal connection has been established between HMAX itself and visual attention.

In this work, we suggest a modification of the HMAX architecture that makes the connection between recognition and visual saliency explicit. We start by investigating the biological plausibility of statistical inference and learning tuned to the statistics of natural images. Building on prior work by [56], we show that a rich family of statistical decision rules, confidence measures, and risk estimates, can be implemented with the computations attributed to the standard neurophysiological model of V1 [75, 21, 69, 20]: a combination of linear filtering, divisive normalization, non-linearities, and spatial pooling. In fact, it is shown that all these computations have precise statistical meaning, contributing to an overall probabilistic interpretation where simple cells compute posterior probabilities and complex cells estimate statistical risks. It follows that a number of statistical operators can be implemented with biological hardware, through simple re-arrangement of lateral divisive connections, non-linearities, and pooling. We next establish a connection to saliency mechanisms, by showing that various proposals for the measurement of visual saliency, from both the biological and computer vision literatures, can be implemented with biologically plausible reconfigurations of the standard neurophysiological model. By replacing the first layer of the HMAX ar-
chitecture with these saliency networks, we conduct a rigorous experimental study of three questions at the intersection of attention and feedforward object recognition: 1) whether saliency benefits visual recognition, 2) whether the gains depend on the type of saliency considered (e.g. top-down vs bottom-up) or even the specific saliency algorithms, and 3) whether max-based pooling has an advantage over the classical linear operator. We note that the goal is not to investigate whether saliency is beneficial as a means to serialize recognition when there are multiple objects within the field of view, as has been done in [111, 168], or whether there are are gains in complementing recognition with an independent saliency path. Instead, we consider the question of whether saliency is intrinsically important for recognition, even when there is a single object in the field of view, as is suggested by computer vision research. Or, in other words, whether in addition to its predominant role within the “where” pathway, saliency also plays a role within the “what” pathway of object recognition. It is shown that the addition of saliency can significantly improve recognition performance, but that this is not independent of the saliency principle adopted. Best results are obtained with top-down saliency mechanisms that equate saliency to classification confidence.

III.B Natural image statistics

Various authors have shown that the empirical distribution of the response $X$ of a band-pass filter to a wide variety of natural imagery is accurately modeled by the generalized Gaussian distribution (GGD) [74, 19, 35]. This distribution is defined as

$$P_X(x; \alpha, \beta) = \frac{\beta}{2\alpha \Gamma(1/\beta)} e^{-\left(\frac{|x|}{\alpha}\right)^\beta} \tag{III.1}$$

where $\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}dt$, $t > 0$ is the Gamma function, $\alpha$ a scale parameter, and $\beta$ a parameter that controls the shape of the distribution.

The parameters $\alpha, \beta$ can be learned in multiple ways, including the method of moments [74], maximum likelihood [35], or Bayesian maximum a posteriori
Figure III.1  Histogram of responses of a set of Gabor filters to a collection of natural images, and its MAP fit by the GGD model with $\beta = .5$.

(MAP) estimation [56]. We adopt the latter, using a (Gamma distributed) conjugate prior for the scale parameter $\alpha$. Given a sample of training observations $D = \{x_1, \ldots, x_n\}$, this leads to [56]

$$\hat{\alpha}_{MAP}^\beta = \frac{1}{\kappa} \left( \sum_{j=1}^{n} |x_j|^\beta + \nu \right), \quad \text{with } \kappa = \frac{n + \eta}{\beta},$$  (III.2)

where $\eta$ and $\nu$ are prior hyper-parameters. The details of the prior are not crucially important, as its role is simply to regularize the feature responses, so as to prevent a null scale estimate. In our implementation we use $\eta = 1$ and $\nu = 10^{-3}$. The MAP estimate of the shape parameter $\beta$ is more complex. However, for natural images this parameter tends to be fairly stable, usually taking values between .5 and .8 [145]. We have found $\beta = .5$ to maximize the likelihood of a large sample of responses of a set of Gabor filters to a random collection of natural images. This is illustrated in Figure Figure III.1, which shows the log-probability histogram of the Gabor responses and the MAP GGD fit for $\beta = .5$. This value was used in all experiments reported in this work.

III.C  Statistical inference

The biological plausibility of probabilistic inference with GGD stimuli was studied in [56]. This work has shown that, for such stimuli, the fundamental
computations of probabilistic inference and learning can be implemented with the
standard computational model of simple and complex cells [75, 21, 69, 20]. In
what follows, we extend the procedures introduced by [56] to show that a much
broader set of computations, summarized in Table III.1, is biologically plausible.
These computations are described in the second column of the table. Although
their biological implementation turns out to be possible with subtle modifications
to the computations of [56], namely the introduction of various non-linearities,
this extension substantially broadens the scope of the underlying computational
framework. For example, the operations now considered are critical to the design
of networks that address top-down problems such as object recognition. In fact,
as will be shown in Section III.G.3, the performance of such top-down networks
can be quite sensitive to the precise choice of statistical inference principle, and
associated non-linearities.

Table III.1 is organized in three sections. The first reports to inference
from a single observation $x$. It starts with the most atomic computation of sta-
tistical inference: the evaluation of the log-probability $\log P_X(x)$ of an observation
$x$. A perceptual system can use this probability to make optimal decisions regard-
ing the classification of $x$ with respect to a target and a null hypothesis. These
are identified by a class label $Y$ that takes the values $Y = 1$ for the target and
$Y = 0$ for the null hypothesis. Optimal decision-making is frequently defined in the
minimum probability of error (MPE) sense, under which the optimal procedure is
the Bayes decision rule [38]. This consists of thresholding the log-likelihood ratio
(LLR)

$$\log \frac{P_{X|Y}(x|1)}{P_{X|Y}(x|0)},$$

and selecting the target hypothesis whenever this ratio is above threshold. An
equivalent implementation of this decision rule is to choose the target hypothesis
when the posterior target probability, $P_Y|X(1|x)$, is above $1/2$. The process is
illustrated in Figure III.2, for an object recognition problem where the target
is the class of airplanes. Given a set of example images from this class, and
Table III.1  Operations of statistical inference under the GGD model.  $\psi(x)$ is defined as $\psi(x) = \frac{1}{2} \log \frac{x}{1-x}$.

<table>
<thead>
<tr>
<th>Definition</th>
<th>Under GGD statistics</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single observation $x$ inference</td>
<td></td>
<td></td>
</tr>
<tr>
<td>negative log-likelihood (NLL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$- \log P_X(x)$</td>
<td>$l^\beta_\alpha(x) = \left( \frac{</td>
<td>x</td>
</tr>
<tr>
<td>log likelihood ratio (LLR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\log \frac{P_{X</td>
<td>Y}(x</td>
<td>1)}{P_{X</td>
</tr>
<tr>
<td>target posterior(TP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_{Y</td>
<td>X}(1</td>
<td>x)$</td>
</tr>
<tr>
<td>information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I(Y; X = x)$</td>
<td>$\xi{\sigma[g(x)]}$</td>
<td>$\xi(x) = \log 2 + x \log x + (1 - x) \log(1 - x)$</td>
</tr>
<tr>
<td>Measures of detection confidence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLRC($x$)</td>
<td>$\tilde{\psi}{P_{Y</td>
<td>X}(1</td>
</tr>
<tr>
<td>IC($x$)</td>
<td>$\tilde{\xi}{P_{Y</td>
<td>X}(1</td>
</tr>
<tr>
<td>expected NLL</td>
<td>$E_X[- \log P_X(x)] = \frac{1}{n} \sum_{i=1}^{n} l^\beta_\alpha(x_i)$</td>
<td>$H[X]$</td>
</tr>
<tr>
<td>expected LLR</td>
<td>$E_X\left[ \log \frac{P_{X</td>
<td>Y}(x</td>
</tr>
<tr>
<td>MI</td>
<td>$E_X[I(Y; X = x)] = \frac{1}{n} \sum_{i=1}^{n} \xi{\sigma[g(x_i)]}$</td>
<td>$I(Y; X)$</td>
</tr>
<tr>
<td>expected confidence (LLR)</td>
<td>$E_X[LLR(x)] = \frac{1}{n} \sum_{i=1}^{n} \psi{\sigma[g(x_i)]}$</td>
<td>$KL[P_{X</td>
</tr>
<tr>
<td>expected confidence (MI)</td>
<td>$E_X[IC(x)] = \frac{1}{n} \sum_{i=1}^{n} \xi{\sigma[g(x_i)]}$</td>
<td>$KL[P_{X</td>
</tr>
</tbody>
</table>
a set of examples from the null hypothesis (in this case any object other than a plane), the visual system relies on a set of bandpass (e.g. Gabor) filters to extract visual features characteristic of the two classes. The GGDs that best fit the distributions, $P_{X|Y}(x|i), i \in \{0, 1\}$, of filter responses under the two hypotheses are then estimated. Given a new image, the corresponding features are extracted, and the LLR of (III.3) is computed, using these GGDs. Thresholding this quantity then produces a binary map that indicates the locations of the target within the visual field.

The LLR is one of various quantities that play an important role in statistical inference and optimal decision making. Table III.1 includes a number of others, which we review in more detail in the remainder of this section. A graphical illustration of these measures, in the context of object recognition, is presented in Figure III.3 and Figure III.4.

An alternative optimality criteria for decision making, commonly referred as infomax [101], is to maximize the information about the class label $Y$. This criterion underlies many classification procedures proposed in the machine learning literature, including logistic regression and some forms of boosting (logitBoost) [68, 51]. Its maximization has also been proposed as a fundamental principle for the organization of perceptual systems [101, 5, 6]. In this case, inference is based on the information

$$I(Y; X = x) = \sum_i P_{Y|X}(i|x) \log \frac{P_{X,Y}(x,i)}{P_X(x)P_Y(i)}$$  (III.4)

that the observation $x$ provides about the class $Y$.

The second section of Table III.1 refers to the evaluation of confidence measures. These complement the decision that $x$ belongs to the target class (target detection), by quantifying how confident the classifier is about this decision. Obviously, the confidence measure should be derived from the principle used for
Figure III.2 Object recognition with the LLR measure. The learning stage is shown at the top of the figure. Gabor filtering is applied to examples of the target and null class. In this example, the target is the class of airplane objects. The probability distributions of the filter responses are then modelled with the GGD distribution. This enables the detection of objects from the target class in previously unseen images, as shown at the bottom. Given the filter responses to an unseen image, and the GGD estimates learned during training, the LLR is computed at each location of the visual field.
Figure III.3  a) an image, b) magnitude of Gabor responses, c) NLL, d) LLR, e) target posterior probability, f) information, g) LLRC(x), h) IC(x). The bars on the side of each image show the range of values corresponding to the pixel amplitudes.
Figure III.4  a) an image, b) magnitude of Gabor responses, c) NLL, d) LLR, e) target posterior probability, f) information, g) LLRC(x), h) IC(x).
inference. This leads to two confidence measures based on the likelihood ratio 

\[ LLRC(x) = \begin{cases} \frac{1}{2} \log \frac{P_{Y|X}(1|x)}{P_{Y|X}(0|x)}, & P_{Y|X}(1|x) \geq .5 \\ 0, & P_{Y|X}(1|x) < .5, \end{cases} \]  

(III.5)

and the information measure 

\[ IC(x) = \begin{cases} I(Y;X = x), & P_{Y|X}(1|x) \geq .5 \\ 0, & P_{Y|X}(1|x) < .5. \end{cases} \]  

(III.6)

An important property is that, in both cases, the confidence measure is “one-sided”, i.e. non-zero only if \( x \) is classified as a target. Although undesirable for the bottom-up problems considered in [56], we will see that this property becomes quite important for success in top-down problems, such as recognition. The two measures can be expressed as a transformation of the posterior target probability \( P_{Y|X}(1|x) \). As indicated in Table III.1, these transformations are

\[ \tilde{\psi}(x) = \begin{cases} \frac{1}{2} \log \frac{x}{1-x}, & x \geq .5 \\ 0, & x < .5 \end{cases} \]  

(III.7)

for \( LLRC(x) \) and 

\[ \tilde{\xi}(x) = \begin{cases} \log 2 + x \log x + (1 - x) \log(1 - x), & x \geq .5 \\ 0, & x < .5 \end{cases} \]  

(III.8)

for \( IC(x) \). They are shown in Figure III.5.

The third section of Table III.1 addresses the characterization of the random variable \( X \). This enables tasks like 1) feature selection, e.g. the identification of the most discriminant Gabor filters for a particular detection problem, or 2) the determination of the entropy of \( X \), e.g. to evaluate the uncertainty of the feature responses. This characterization usually requires the computation of empirical averages of the statistical inference operators discussed above, from a sample of observations \( R = \{x_1, \ldots, x_n\} \). Such averages are empirical estimates of popular statistical risks, which are referenced in the right-most column. These include the entropy

\[ H[X] = - \int P_X(x) \log P_X(x) dx = E_X[- \log P_X(x)], \]  

(III.9)
Figure III.5 Non linear transformations of the posterior target probability that produce the information $I(Y; X = x)$ (left) and the confidence measures LLRC(x) (center) and IC(x) (right).

the mutual information

$$I(Y; X) = \sum_{i} \int P_{X,Y}(x, i) \log \frac{P_{X,Y}(x, i)}{P_{X}(x)P_{Y}(i)} \, dx = E_{X}[I(Y; X = x)]$$  \hspace{1cm} (III.10)

or the Kullback-Leibler (KL) divergence

$$KL[P_X(x)||Q_X(x)] = \int P_X(x) \log \frac{P_X(x)}{Q_X(x)} \, dx = E_X \left[ \log \frac{P_X(x)}{Q_X(x)} \right].$$  \hspace{1cm} (III.11)

Once again, each inference principle leads to a different risk. For example, the expected LLR is a difference of two KL divergences

$$KL[P_X(x)||P_{X|Y}(x|0)] - KL[P_X(x)||P_{X|Y}(x|1)],$$

while the expected value of the information measure $I(Y; X = x)$ is the mutual information $I(Y; X)$ between the observation $X$ and the class label $Y$. Finally, it is also possible to rely on expectations of the confidence measures of (III.5)-(III.6). These can be seen as one-sided versions of the KL difference and mutual information, which only average sample points identified as belonging to the target class (by the Bayes decision rule). Such averaging is equivalent to computing expectations with respect to the target class conditional distribution $P_{Y|X}(x|1)$, rather than $P_X(x)$. It, for example, simplifies the KL difference of (III.12) into the more standard KL divergence

$$KL[P_{X|Y}(x|1)||P_{X|Y}(x|0)] = \int P_{X|Y}(x|1) \log \frac{P_{X|Y}(x|1)}{P_{X|Y}(x|0)} \, dx,$$

\hspace{1cm} (III.13)
Again, their one-sided nature makes these risks particularly effective for top-down problems, such as target detection or recognition.

All risks based on KL divergences or mutual informations measure the discriminant power of $X$ for target detection, and can be used for feature selection. When $X = \{X_1, \ldots, X_k\}$ is a set of bandpass features, the dependencies of the feature responses to natural images tend to carry little information about the class label [158]. This can be exploited to simplify the joint mutual information of the features with the class label into

$$I(X; Y) \approx \sum_k I(X_k; Y), \quad \text{(III.14)}$$

and justifies the computation of the overall discriminant power of $X$ by adding the discrimination measures derived from each feature channel. We use this procedure to integrate the empirical risks of Table III.1 across feature channels.

### III.D Inference under the GGD

When $X$ follows a GGD, the computations above can be simplified into the form shown in the third column of Table III.1. Here, all equations assume that $X$ is either a GGD random variable of parameters $(\alpha, \beta)$, or a GGD random variable when conditioned on the class $Y$. In this case, the class conditionals $P_{X|Y}(x|i)$ have parameters $(\alpha_i, \beta_i), i \in \{0, 1\}$. It is also assumed that $P_Y(0) = P_Y(1) = 1/2$, but this could be generalized into any label distribution. As noted by [56], the form of the negative log-likelihood

$$l_\alpha(x) = \frac{|x|^\beta}{\alpha^\beta} + K \quad \text{(III.15)}$$

is a straightforward consequence of (III.1). It follows that large values of $|x|$ indicate the locations of visual stimuli of low probability within the field of view. This is illustrated in Figure III.3 and Figure III.4 a)-c), which present two images, the magnitude $|x|$ of their convolution with a Gabor filter, and the NLL $l_\alpha^\beta(x)$ for the MAP GGD fit with $\beta = .5$. Note that the latter emphasizes details of the
object or background which have very distinctive appearance from the rest of the image. In this sense, the log-likelihood operator behaves as an interest point operator, similar to a number of interest point operators currently popular in computer vision [66, 112, 137, 83].

By definition, the LLR is a difference of two negative log likelihoods. It can be written as

\[ g(x) = \log \frac{P_{X|Y}(x|1)}{P_{X|Y}(x|0)} = \left( \frac{|x|}{\alpha_0} \right)^\beta - \left( \frac{|x|}{\alpha_1} \right)^\beta + T, \]  

(III.16)

where \( T = \log \left( \frac{\alpha_0}{\alpha_1} \right) \). Figure III.3-Figure III.4 d) show the LLR for motorbike detection on the images of a). In both cases, \( \alpha_1 \) was learned from a collection of bike images (target hypothesis), and \( \alpha_0 \) from a random collection of natural images (null hypothesis). The LLR emphasizes the region of the motorbike, which is approximately uniformly highlighted, and inhibits the background.

Simple application of Bayes rule leads to the well known relation

\[ P_{Y|X}(1|x) = \frac{1}{1 + \frac{P_{X|Y}(x|0)}{P_{X|Y}(x|1)}} = \sigma \left[ g(x) \right] \]

where

\[ \sigma(x) = 1/(1 + e^{-x}), \]  

(III.17)

is the sigmoid function. Hence, the target posterior is a sigmoidal transformation of the LLR. Similarly, (III.4) can be written as \( I(Y; X = x) = \xi[P_{Y|X}(1|x)] \), with the non-linearity

\[ \xi(x) = \log 2 + x \log x + (1 - x) \log(1 - x) \]

shown in Figure III.5. The application of these non-linearities to the images of Figure III.3-Figure III.4 d) are shown in Figure III.3-Figure III.4 e)-f). They remap the LLR into the range \([0 - 1]\). While [56] have combined \( \sigma(x) \) and \( \xi(x) \) into a single non-linearity, there are non-trivial benefits in decoupling the two components. Note, in particular, that while the sigmoidal transformation maintains the emphasis on the bike region, the non-linearity associated with the information measure
re-emphasizes some of the background. This is due to the fact that the latter is insensitive to the sign of the LLR (or, equivalently, to the sign of $P_{Y|X}(1|x) - 1/2$).

In a strict information theoretic sense, the absence of an object is as informative as its presence for object detection (the classifier is simply very confident in the assignment of the image pixels to the background class). This is, however, undesirable for object detection, where the role is to detect object, and not background.

When the two non-linearities are decoupled, this problem can be corrected by resorting to the measures of classification confidence of (III.5) and (III.6), which can be computed by composition of the sigmoid with the non-linearities of (III.7) and (III.8). The result, shown in Figure III.3-Figure III.4 g)-h) is a strong suppression of regions that belong to the background. This suppression enables very non-trivial gains in recognition accuracy, as will be shown in Section III.G.3. Finally, all empirical risks can be computed by averaging some combination of these non-linearities. In summary, as noted in Table III.1, most operations of statistical inference with GGD stimuli are non-linear mappings of the LLR $g(x)$ of (III.16).

### III.E Biological plausibility

[56] have shown that, given a sample $R$ from a GGD distribution and using the estimate of (III.2) in (III.15),

$$l^\beta(x) = \kappa \frac{|x|^\beta}{\sum_j |x_j|^\beta + \nu} + K. \quad (III.18)$$

The absolute value of $x$ can be computed by half-wave rectification, i.e. as $|x| = x^+ + x^-$ where $x^+ = \max(x, 0)$ and $x^- = \max(-x, 0)$. This leads to the sequence of computations attributed to simple cells by the standard neurophysiological model of V1 [75, 21, 69, 20]: linear filtering to produce a filter response $x$, half-wave rectification, and divisive normalization by the responses of other cells. For simplicity, we omit the decomposition into the rectified components $(x^+, x^-)$ from all equations and network diagrams, working with $|x|$ instead. The combination of absolute value and divisive normalization as in (III.18) has recently been found
Figure III.6 The NLL is computed by a simple cell that normalizes a feature response $x$ by the responses of its spatially neighboring units.

to substantially improve the recognition accuracy of classical convolutional networks [80, 123, 124]. However, no principled justification has been given for the importance of these operations. The discussion above suggests that this importance follows from their interpretation as estimators of the fundamental quantity of statistical inference (log-probability). The network representation of the simple cell is shown in Figure III.6.

Since the LLR is the difference of two log-probabilities, given two samples $\mathcal{R}_0$ and $\mathcal{R}_1$ from the null and target class, respectively, it follows that

$$g(x) = \frac{1}{\kappa} \sum_{x_j \in \mathcal{R}_0} |x_j|^\beta + \nu - \frac{1}{\kappa} \sum_{x_j \in \mathcal{R}_1} |x_j|^\beta + \nu + T. \quad (III.19)$$

This leads to the biologically plausible implementation of the LLR with the network of Figure III.7.

The main difference with respect to the network of Figure III.6 is that the filter responses are now differentially normalized by the units in the two dashed boxes. These boxes pool the response of other cells in a region $T$ where the training sample $\mathcal{R}$ is collected. The bottom (top) units collect positive (negative) examples, producing an estimate of the GGD scale for the target class (null hypothesis). The region $T$ localizes the cell computations. If $T$ is the entire field of view, the GGD models are average distributions for the feature responses across the latter. For
Figure III.7 A LLR unit divisively normalizes a feature response $x$ differentially, using the outputs of two units that estimate GGD parameters under the target and null hypothesis. With the inclusion of the output non-linearity $\sigma(\cdot)$, this unit computes posterior target probabilities.

smaller $T$, the cell response is tuned to the statistics of a sub-region of the field of view. Hence, the LLR can be computed by a differentially normalized simple cell. This prompted [56] to propose the LLR network as a model for simple cells. There are, however, two significant advantages in further including a sigmoidal non-linearity at the network output, as is now proposed in Figure III.7. First, this turns the cell into an estimator of the posterior target probability $P_{Y|X}(1|x)$, a more central quantity to the computations of Bayesian decision theory than the LLR. Second, it strengthens the biological plausibility of the simple cell model, by accounting for the saturation effects that are well known to hold for simple cell outputs, but are not replicated by the LLR.

Most risk estimates in the lower third of Table III.1 consist of pooling some nonlinear transformation of the posteriors $\sigma[g(x_i)]$, within some region $R$ of the field of view. This makes the associated computations good candidates for complex cells. An example is the MI, for which the pooling operation is represented
III.F Saliency

A number of proposals for the measurement of visual saliency can be implemented by the networks of Table III.1. We consider two bottom-up saliency methods, based on the detection of rare features, and a top-down approach, discriminant saliency, which accounts for the classes of the objects to detect.

Detection of rare features: a number of authors have advocated the detection of features of low probability as a criterion for visual saliency [130, 180, 18]. As discussed above, this criterion can be implemented with the NLL unit of
Figure III.6. The detection of low probability features is also closely related to the most popular strategy for the detection of interest points in computer vision. A number of detectors from this literature identify image structure such as corners [66], locations of strong image derivatives [112], wavelet coefficients of large magnitude [137], or local maxima of image entropy [83] that have low probability of occurrence. The features that elicit a strong response by NLL units generalize all these types of structure. For example (see Table III.1), the combination of NLL units with a complex cell that pools its afferents linearly measures the entropy of the underlying feature responses.

It should be noted, however, that NLL units are technically not feature detectors, since they only compute the likelihood of feature responses. One possibility to transform them into detectors is to consider a discriminant version, that tests two hypotheses. Under the null hypothesis, $x$ follows a GGD distribution $P_X(x)$ of parameters $(\alpha, \beta)$ estimated from the visual field. Under the alternative hypothesis, $x$ follows a non-informative distribution $P_X(x) \propto 1$. The likelihood ratio is $g(x) \propto -\log P_X(x)$ and the posterior $P_{Y|X}(1|x) = \sigma(-\log P_X(x)) = \sigma(|x|/\alpha + K)$. The null hypothesis is rejected when $|x|/\alpha$ is large, i.e. large responses are better explained by the non-informative distribution. This implies that such responses are rare within the field of view. From an implementation point of view, the discriminant unit is identical to the NLL of Figure III.6, with the addition of an output sigmoid. We denote this combination as a rare feature detector (RFD).

**Discriminant saliency:** Discriminant saliency is defined with respect to a target and a null hypothesis. In the object detection context, the target is the class of objects to detect while the null hypotheses encompasses all stimuli outside that class. Locations of the visual field that can be assigned to the target class with minimal probability of error are declared salient, with degree of saliency equal to the classification confidence [58, 56, 57]

$$S(x) = \begin{cases} I(Y; X = x) & \text{if } P_{Y|X}(1|x) > .5 \\ 0, & \text{otherwise} \end{cases}$$

(III.20)
This is the $IC(x)$ measure of Table III.1. If multiple responses $\{x_1, \ldots, x_K\}$ from feature $X$ are available, the saliency of $X$ is defined as $I(X; Y) = \frac{1}{K} \sum_i S(x_i)$, i.e. the expected confidence (MI) measure of the table. Saliency measurements derived from multiple feature channels are combined with (III.14). The last third of Table III.1 suggests a number of other discriminant possibilities for measuring feature saliency: KL difference, mutual information $I(X; Y)$, or KL divergence. These measures differ from the expected confidence (MI), adopted by discriminant saliency, in relatively small details (mostly non-linearities). Such details could nevertheless be of consequence. For example, [80] has found that simply taking the absolute value of the output of each unit of a classical convolutional network can produce drastic improvements in its recognition accuracy. The discussion above shows that these details can also completely alter the semantics of the network computations. For example, unlike the expected confidence (MI), the MI does not emphasize feature presence and could identify as salient a feature that is always absent from the target class. This is desirable for bottom-up saliency [56] but not necessarily for top-down applications, such as object detection or localization. We evaluate the performance of these measures in the following section, where it is shown that the choice of non-linearities can indeed have a significant impact on recognition performance.

\section*{III.G Results}

HMAX networks emulate the organization of the visual system by a cascade of two layers of simple and complex cells. We investigated the role of saliency in recognition by replacing the first HMAX layer with a saliency network. Under HMAX, this layer is quite simple: simple units perform filtering, and complex units pool simple unit responses within a spatial neighborhood, using a maximum operator. While these simple units have no probabilistic interpretation, max-based complex units are an interesting alternative to the sample averages of Table III.1.
They act more like a feature selection mechanism: rather than averaging responses, max-based pooling identifies the location of most salient response. This appears natural for detection-based saliency measures, e.g. the RFD. By replacing the first HMAX layer with a saliency network we can thus investigate three questions:

1. is saliency important for visual recognition?

2. how do the various saliency criteria compare on an objective task, such as object recognition?

3. is there an advantage in using max versus the classical linear pooling?

In the broader neural network literature, there have been recent showings that some details of the network computations, e.g. what type of non-linearities or normalization is performed, can have a substantial impact in recognition accuracy [80, 124]. As discussed above, the statistical interpretation of these operations makes it possible to assign semantics to all computations, with respect to optimality principles for discrimination, statistical inference, measurement of information, etc. This enables a more efficient search for optimal computations than trial-and-error [80], or brute-force optimization [124]. To study these questions we performed a number of experiments, which are discussed in the remainder of this section.

III.G.1 Experimental set up

We start with a simple synthetic problem that provides intuition on the benefits of top-down discriminant saliency for recognition, and then present more extensive experiments on the Caltech101 benchmark, commonly used to evaluate object recognition performance. All experiments were based on the HMAX network, whose first layer was replaced by a saliency network. On Caltech101 we tested all saliency measures in the lower third of Table III.1, as well as RFD, and the saliency detector of [78]. For completeness, we also evaluated the use of a classical sigmoidal layer (no complex units or pooling, simple units a combination of filtering and a sigmoid) in the first HMAX layer, and the HMAX network itself. To
investigate the advantages of max over linear pooling, all saliency networks were implemented with both. On the synthetic experiment we compared an HMAX network, HMAX with first layer replaced by a bottom-up saliency network of RFD units (HMAX+RFD), and HMAX with first layer replaced by a top-down saliency network of expected confidence (LLR) units (HMAX+EC).

In all experiments, for saliency units that involve divisive normalization, the pooling region $\mathcal{T}$ of the normalizing units was the whole image. In the case of bottom-up saliency (NLL or RFD units) the normalization is performed on-line, i.e. dividing by neighboring responses to the image to recognize. For top-down saliency (LLR units) the normalizing coefficients are learned during training, when the network is exposed to images from the target and null hypotheses. For complex units, the pooling region $\mathcal{R}$ was as specified in [117].

The second layer of the HMAX network consists of a set of radial basis function (RBF) units, centered at prototypes randomly sampled from the responses of the first HMAX layer, during training. On Caltech101 we used the implementation of [138], which includes 4,075 RBF units. On the synthetic experiment we used a smaller network of 100 units. For LLR units, training produces two divisive normalization parameters ($\alpha, \beta$) per object class. For a given RBF prototype $P$, the parameters of the afferent simple units are set to the values $\alpha, \beta(P)$ with which $P$ was learned (i.e. the parameters learned from the image class which originated $P$). Other than these modifications, the network is exactly as described in [117].

III.G.2 Synthetic problem

To gain some insight on the role of discriminant saliency in recognition, we considered the simple problem of learning to differentiate underlined from non-underlined characters. This was formulated as a two-class recognition problem, involving the stimuli of Figure III.9.

Each network was trained with the top two images of the figure, using underlined Xs as examples from the target class, and regular Xs as example non-
Figure III.9 Detection of underlined characters. Top row: training examples from target and non-target class. Bottom rows: examples of test stimuli from the target and non-target class, and layer 1 responses from the three networks considered.

targets. This made the classes identical up to a salient feature of the underlining concept (the underline bar). The network was then used to classify 20 test images, containing either targets or non-targets. To increase the difficulty of the task, the character used on the test images (Y) was different from that used for training (X), and random noise was added to all images.

The recognition accuracy achieved by the three networks was 90% for HMAX+EC, 55% for HMAX+RFD, and 50% for HMAX. The superior performance of the network with top-down saliency can be understood by analyzing the intermediate network responses, shown in Figure III.9 and Figure III.10. Consider the response of the first network layer, shown in Figure III.9. The HMAX network only has access to Gabor filter responses, which are very similar for target and non-target. This makes it very difficult for the subsequent HMAX stages to distinguish between the two classes. Because none of the parts of the underlined
Figure III.10  Top: most discriminant filter (the 4 orientation channels are shown) of the second network layer, for HMAX (left) and HMAX+EC (right). By most discriminant it is meant that this is the filter given larger weight by the linear SVM classifier at the network output. Bottom: example output of the simple cells in layer 2, to target and non-target stimuli.

Xs pop-out within the target displays, the saliency response of RFD is basically a contrast enhanced version of the filter responses. This does not improve the recognition accuracy substantially, since contrast variability is not the reason for the poor performance of HMAX on this classification problem (although it can be a source of concern for problems involving natural images where, as we will see in the next section, HMAX+RFD tends to outperform HMAX). Hence, the performance of HMAX and HMAX+RFD is basically identical.

The underline bar is, however, salient in the top-down sense, since it is the only part that distinguishes the target and non-target examples. Because the units of the HMAX+EC network compute the LLR between target and non-target hypothesis, they produce a strong response to underline bars (plausible under target, but not plausible under the non-target hypothesis) and a weak response to everything else (equally plausible, or non-plausible, under the two hypotheses). The network has thus learned that horizontal bars are discriminant features for the detection of underlined characters, and thus salient. Its first layer acts as a detector of these bars, and its very different responses to targets and non-targets
are easily detected by the subsequent network stages. Figure III.10 presents the most discriminant filter of the second layer (4 orientation channels shown), for the HMAX and HMAX+EC networks. Note how the filter of HMAX+EC is a detector of horizontal bars, a property that does not hold for the other networks. In result, the output of the second layer of HMAX+EC is uniformly large for underlined characters, and almost null for non-targets. This is unlike the other two networks, whose second layers respond to both targets and non-targets. It is thus not surprising that HMAX+EC achieves a substantially higher recognition accuracy.

### III.G.3 Caltech101 experiments

To evaluate the impact of the various saliency principles on the classification of natural images, we performed a number of experiments on Caltech101. All experiments were based on the experimental protocol of [117]. We considered the multiclass recognition task, where 30 images per class are used for training and a maximum 50 of the remaining for test. In all experiments the reported recognition rate is the average over 5 independent runs, with different train and test sets (randomly sampled images). Table III.2 presents the recognition accuracy achieved with each variant of the first network layer. A graphic display of these rates, as well as the associated error bars, is shown in Figure III.11.

A few interesting observations can be made. First, the two expected confidence criteria achieve the best results. Their performance is similar, but EC(LLR) attains slightly higher recognition rates. These methods can be implemented with simple units that compute the target posterior probability, i.e. a combination of a differentially and divisively normalized (LLR) unit and a sigmoid $\sigma(.)$. The gains with respect to the remaining networks can be very significant. Second, saliency criteria based on rare features (ENLL and RFD) perform worse than saliency criteria based on discrimination (the expected confidence measures). On the other hand, both rare feature criteria have clearly better performance than sigmoid or
Table III.2 Recognition rates on Caltech101, using 30 training examples per class. All abbreviations are the same as in Table III.1. Furthermore, EC means expected confidence, ELLR expected LLR, ENLL expected NLL, RFD rare feature detection.

<table>
<thead>
<tr>
<th>network</th>
<th>Simple units</th>
<th>Complex units</th>
<th>pooling</th>
<th>acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>div. norm.</td>
<td>NLL</td>
<td>LLR</td>
<td>σ(·)</td>
</tr>
<tr>
<td>EC(LLR)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EC(MI)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ELLR</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ENLL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RFD</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Itti(1998)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HMAX</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

HMAX. This suggests that rare feature (interest point) detection can be useful when statistics of the target object class are not available. Note that, under the rare feature criteria, none of the two network layers requires class-specific training. While the same holds for the saliency detector of [78], its performance (51.8%) is substantially weaker than those of ENLL or RFD.

Third, the “one-sided” confidence measures EC(LLR) and EC(MI) perform substantially better than their “two sided” counterparts, such as the ELLR or the MI used in [56]. This implies that the choice of non-linearities (e.g. ̂ξ instead of ξ or ̂ψ instead of ψ) can have a very non-trivial impact in recognition accuracy.
Figure III.11 Recognition rates on Caltech101, using 30 training examples per class.

It appears to be particularly important for the cells to fire only when the target is present. Fourth, for most networks, max-based pooling has inferior performance to averaging. This implies that it is important to fully characterize features, and not only select locations where they are informative for the classification. The only network for which max pooling consistently achieves better performance is HMAX (where the lack of sophistication of the simple units makes the network with average pooling linear). Furthermore, max-based pooling is prone to large performance variability. For example, the EC(MI) network drops from 60 to 53% recognition rate when averaging is replaced by max pooling. Finally, the classical sigmoid layer has the worst performance of all considered. However, the simple addition of a pooling stage can improve performance considerably, especially when combined with max pooling.

III.G.4 Comparison to state-of-the-art results

To the best of our knowledge, the current state-of-the-art results for object recognition with HMAX networks are those presented in [117]. This work
Table III.3  Multiclass classification results for 101 categories. The table shows the improvement by changing/adding components to base model.

<table>
<thead>
<tr>
<th>Model</th>
<th>15/cat</th>
<th>30/cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>base model of [138]</td>
<td>33</td>
<td>42</td>
</tr>
<tr>
<td>+ sparse S2 inputs [117]</td>
<td>35</td>
<td>45</td>
</tr>
<tr>
<td>+ inhibited S1/C1 outputs [117]</td>
<td>40</td>
<td>49</td>
</tr>
<tr>
<td>+ limited C2 invariance [117]</td>
<td>48</td>
<td>54</td>
</tr>
<tr>
<td>+ feature selection [117]</td>
<td>51</td>
<td>56</td>
</tr>
<tr>
<td>EC(LLR) with sum pooling + feature selection described in [117]</td>
<td>56</td>
<td>62</td>
</tr>
<tr>
<td>convolutional net of [123]</td>
<td>-</td>
<td>42</td>
</tr>
<tr>
<td>+ second HMAX layer</td>
<td>-</td>
<td>56</td>
</tr>
<tr>
<td>convolutional net of [80]</td>
<td>-</td>
<td>56</td>
</tr>
<tr>
<td>+ random filters</td>
<td>-</td>
<td>63</td>
</tr>
<tr>
<td>+ unsupervised filters</td>
<td>-</td>
<td>64</td>
</tr>
<tr>
<td>+ backpropagation filters</td>
<td>-</td>
<td>66</td>
</tr>
<tr>
<td>[93]</td>
<td>56</td>
<td>65</td>
</tr>
<tr>
<td>[177]</td>
<td>59</td>
<td>66</td>
</tr>
</tbody>
</table>

reported significant improvements over the base HMAX performance, through a number of enhancements to the original network. Some of these involved additional training, e.g. to select features, others are heuristics that were shown to improve performance. Table III.3 presents the contributions by these enhancements, as reported in [117].

As can be seen from the table, the simple use of the saliency layer, without any further optimization, outperforms the gains of all enhancements of [117]. One of these improvements is a feature selection stage. Rather than using 4,075 randomly sampled prototypes, a larger set of 12,000 are collected. The network is trained with this larger set, and a support vector machine is used to select the most discriminant 4,075. When we retrained the network containing the saliency layer in this manner, the performance increased to 64%, as opposed to the 56% reported by Mutch and Lowe. While we have not yet experimented with any of their other suggestions, or performed any other optimization, these results suggest that the inclusion of saliency can significantly boost the performance of feedforward object
recognition.

In the broader area of convolutional networks, recent studies have addressed the role of non-linearities and normalization in object recognition [123, 80]. These works advocate the use of divisive normalization as a form of contrast normalization, that improves the robustness of the neural network when trained from small samples, as is the case of Caltech101 [80]. This is a strictly bottom-up explanation for the role of divisive normalization, and comparable to the ENLL and RFD saliency measures discussed in this work. Comparison with these methods should be performed with care, since the network parameters are not the same. For example, while it has become somewhat popular to claim that method of [123] beats the state-of-the-art in computer vision, the truth is that its implementation is far from the standard in this area. For example, while (for computational efficiency) most computer vision implementations rely on a relatively small set of filters (e.g. Gabor filters at 4 orientations) and a relatively small number of network outputs (4075 for the first HMAX network [138], 12000 for enhanced HMAX [117]), this method relies on a much larger filter set (12 orientations), and a much larger output dimensionality (86400-116400). The network has a single layer and is complemented by a classifier that combines a principal components analysis of very disputable biological plausibility, and an SVM. While the recognition accuracy originally reported by the authors is of 65% (30 images per category), our implementation with 1) the Gabor filter front-end and 2) the output dimensionality used by the HMAX networks only achieved 42%. Further inclusion of the second HMAX layer raised recognition performance to 56%. We note that this is consistent with the results of Table III.2, as the network of [123] is similar to the RFD network. Hence, it is not surprising that the results are in between those of ENLL (58.2%) and RFD (55.1%).

Similar performance was documented by [80], who have obtained 55.8% accuracy with a two layer network including divisive normalization in the two layers (as opposed the one we tested, where only the first layer was modified).
This work has tested a number of extensions, including the use of filters learned from the training data, in both a bottom-up and top-down manner. All results reported are lower than those achieved with the EC(LLR) network, except when the filters are trained in a discriminant manner. Note that, in this case, the convolutional network has two layers of trained filters and divisive normalization, network training is orders of magnitude more complex than that required by the saliency network (back-propagation for the former vs. the individual tuning of the divisive normalization weights of each simple cell, according to (III.2), for the latter), and the gains are very marginal (65.5% vs 64%). The filters of the EC(LLR) network could also have been trained in a discriminant manner, but we have not attempted to perform this optimization.

For completeness, we also report the state-of-the-art results on Caltech101 from the broader recognition literature in computer vision, where biological plausibility is not a constraint. We consider here only methods that use a single image representation, and are therefore comparable to the networks proposed above. In this class, the best performance in the literature is in the range of 65−66% [93, 177] and barely superior to the 64% now reported for the biologically plausible networks. Obviously, better performance should be attainable by combining multiple image representations, e.g. by adding features that capture color or shape properties to the set of Gabor functions that we consider in this work. This is indeed a popular strategy in the computer vision literature, where it has been shown that substantial improvements over [93, 177] can be achieved with support vector machines combining multiple kernels [157, 60]. Such combinations of multiple image representations could also be applied to the networks that we have proposed, but are beyond the scope of this work.

III.H Discussion and Conclusion

Overall, the results presented above support three main conclusions:
• saliency (attention) has a significant positive impact on recognition,

• but this impact is largest when saliency is discriminant (of a top-down nature). Unsupervised learning of interest points does not perform as well, although it consistently achieves better performance than no saliency at all (standard HMAX);

• max-based pooling does not appear to have an advantage over averaging, indicating that selecting discriminant features is more important than locating them exactly.

It could be argued that replacing the raw filter outputs with discriminant saliency measures is simply a form of normalization, whose benefits have already been pointed out in the literature [123, 80]. While normalization has advantages of its own, as shown by the gains of the ENLL and RFD networks over their sigmoidal counterpart, this is not the whole story. The results above show that non-trivial additional gains can be obtained with intelligent normalization, which tunes the cell responses according to the target recognition class, at a very marginal cost in computation. This is a top-down saliency operation.

To illustrate the benefits of this type of saliency for classification of natural images, we examined the intermediate computations of the different types of networks. Figure III.12 shows the output of the saliency layer for an example image of the “accordion” class.

The figure shows the saliency maps produced for four Gabor orientation channels. The first row presents the magnitude $|x|$ of the Gabor responses (no saliency processing), the second row the output of the NLL units (bottom-up processing), and the third row that of the LLR units trained for accordion detection (top-down saliency). Note that both types of saliency units reinforce the contrast of certain areas of the image, leading to a more distinctive visual signature than the simple magnitude of Gabor responses. The responses of the two types of saliency units are, nevertheless, quite different. NLL has no knowledge of the accordion
Figure III.12 An image from the “accordion” class, and corresponding saliency outputs for Gabor channels of four orientations. Top row: magnitude of the Gabor responses. Center: saliency maps produced by NLL units. Bottom: saliency maps of LLR units.
class, and simply highlights visual features that have low probability within the field of view. These tend to be the keyboards that appear on each side of the instrument. The diagonal edges, which are a distinctive pattern of the accordion object but plentiful on this image, are suppressed. This implies that there is some loss of information, a limitation of bottom-up saliency for recognition: universal saliency criteria (such as low probability, or contrast normalization) fail to capture the salient attributes that are specific to any given object class.

The top-down LLR units exhibit a substantially different behavior. For orientation channels that do not contain substantial discriminant information about the target class, they behave similarly to NLL units. However, for orientations that capture distinctive object patterns (such as the large density of parallel lines in the accordion class), they respond very strongly throughout the field of view, highlighting the whole object. The resulting saliency patterns are thus much more distinctive templates than those produced by Gabor filtering or NLL. When used in the second HMAX layer, these templates are much more discriminant for the target class, enabling better detection performance. In summary, the attributes that are salient for object recognition vary from one object class to another. The identification of such attributes requires top-down processing informed by the class structure associated with the recognition problem. Discriminant saliency implements this type of processing, leading to the extraction of intermediate features that are highly informative for object recognition. This results in higher recognition rates.

We finish by emphasizing one of the most interesting findings of this work: that subtle modifications to the computations of [56] can lead to substantial changes of network behavior. These include 1) obtaining good performance on top-down tasks such as recognition (rather than just bottom up saliency), 2) computing new statistical quantities of interest, namely all measures of Table III.1, 3) explaining properties such as simple cell saturation, and 4) assigning semantics to all network components. All of these help understand why network modifications
that appear minor a-priori can have a dramatic impact in performance. For example, while [56] have shown that, among the non-linearities of Figure III.5, $\xi(x)$ performs best for bottom-up saliency, the results now presented show that $\tilde{\xi}(x)$ is clearly better for top-down saliency. This can be seen from Table III.2, where replacing $\tilde{\xi}(x)$ by $\xi(x)$ leads to a substantial decrease of recognition accuracy, e.g. from 60.3% (EC(MI)) to 50.3% (MI).

Although the dramatic influence of non-linearities on recognition performance has been documented in the literature [80], it can be quite puzzling in the absence of the statistical interpretation now provided. Why would simply changing a non-linearity degrade the performance so much? And why does it matter so much that the non-linearity is “one sided”? The statistical interpretation clarifies this behavior: while the EC(MI) is a detector of target presence, the MI is equally happy to detect target presence or absence. The semantics of the network computations are, therefore, completely different. Under MI, the network produces large responses to background regions that can be classified as either target or non-target with high confidence. This increases the difficulty of target detection. On the other hand, under EC(MI) the network only produces large responses to regions that contain the target.

### III.I Acknowledgments

The text of Chapter III, in full, is based on the material as it appears in: Sunhyoung Han and Nuno Vasconcelos, “Biologically Plausible Saliency Mechanisms Improve Feedforward Object Recognition”, published in *Vision Research*, 2010. The dissertation author was a primary researcher and an author of the cited material.
Chapter IV

Hierarchical Discriminant
Saliency Network
IV.A Introduction

In neurophysiology, it has been known since the days of Hubel and Wiesel, that cortical networks consist of layers of at least two types of cells, commonly denoted simple and complex [75]. Simple cells are responsible for the linear filtering that tunes cells for certain orientations, and complex cells pool information from multiple simple cells, producing a more invariant representation. This enables the cell responses to be both selective, for certain types of stimuli, and invariant. Over the last decades, a much more complete picture of cell computations has emerged, leading to what is now referred as the standard neurophysiological model [20]. Under this model, simple cells implement a combination of filtering, rectification, and divisive contrast normalization, in addition to the classical sigmoidal non-linearity, while complex cells combine pooling with certain non-linearities. Cortical layers are then organized hierarchically. While the receptive fields of cells at the lower hierarchical levels resemble Gabor filters of limited spatial extent, cells at the higher layers have much more complex receptive fields, and pool information from much larger regions of support [122, 125]. This makes them both more selective and more invariant than their low-level counterparts.

These observations have motivated a recent resurgence of interest in network models within computer vision. Prominent examples include the HMAX model of [129, 138, 117] and the convolutional network extensions of [123, 80]. All of them have greater biologically plausibility than classical sigmoidal feedforward networks, and better performance. Extensive experiments have shown that accounting for simple and complex cells [138], including different types of normalization and rectification [80], or optimizing the sequence in which these operations are performed can lead to significant recognition improvements [124]. Unfortunately, there is not much guidance on how to find the best network configuration for a given recognition problem and cannot explain the effect of discriminative training [9] which is believed to be critical in object recognition [61, 141, 121].
Unlike classical sigmoidal networks, where each unit only requires the tuning of a weight vector, the enhanced models require optimization of the network structure itself, namely the sequence and the parameters of the operations performed by each network unit. The complexity of this problem is compounded by the fact that the newer models are not derived from a clear optimality principle, and their components lack a well-defined functional justification. It is not clear what functionality the intermediate layers are serving, why operations such as rectification or contrast normalization make a difference, what justifies the proposed sequences of these operations, or what such sequences are optimal for. In practice, this issue is commonly ignored. In some cases, e.g. HMAX, a particular implementation is selected without much concern about optimality. In other cases, e.g. [80], a few alternative configurations are proposed, and their performance compared on standard object recognition benchmarks. While performance improves, it is difficult to ascertain if the same results would hold on other datasets or whether some untested configuration would achieve better performance. Finally, there have been proposals for an exhaustive search over the set of network configurations [124]. This is, however, infeasible for most applications, tends to produce networks that lack a clear structure, and have an even less clear functional justification than their sub-optimal counterparts.

One consequence of the absence of strong functional justification, is that it is difficult to relate these models to the psychophysics of object recognition. In this area, it is well known that recognition is heavily dependent on the mechanisms of visual attention. Two attentional mechanisms play a role in the recognition process. The first is a preattentive layer driven by simple filters, e.g. Gabor functions. The second is an attentive layer of cells with more complex receptive fields [89]. The simple features of preattentive processing are computed rapidly, in parallel, across the visual field. Attentive processing is slower, more serial, and restricted to a spatial neighborhood of the latter. For example, studies by Thorpe et al. [148] have shown that humans can categorize natural scenes with high ac-
accuracy (close to 94%) and very quickly (in less than 150ms). The fact that such recognition speeds leave no room for propagation of feedback across cortical areas, suggests a purely feedforward process. In fact, given the time required to propagate information through a neuron (about 40ms) these networks are likely to be quite shallow. More recently, Johnson and Olshausen [81] have shown that recognition performance on higher level cued-target tasks has more variable latency, adding between 20 and 170ms to the processing time of the purely feedforward mechanisms. One possible explanation for this is the existence of a top-down template model which is combined with the purely feedforward model in cued-target recognition tasks. Other researches showed that fast response and rapid attention is captured by high-level stimuli [88, 70, 71, 131, 144] and cell response strength depends on the direction in which attention lies relative to the receptive field meaning information about the positional relationship between stimulus is exploited [28].

These observations have motivated the addition to the network models of complex features selective for location, which implicitly encode a focus of attention mechanism. For example, it has been shown [117] that even the addition of rough spatial selectivity to the units in the higher network layers can substantially improve the recognition performance of models such as HMAX. However, the connection between recognition and attention established by such heuristics is quite weak. Much stronger connections are available in the saliency literature, where several models of top-down saliency tie the optimality of saliency judgements to object recognition performance. Examples include discriminant saliency [57], which equates saliency to optimal decision-making, and the guided search [172], biased competition [34] and optimal gains [118] models, where feature maps are globally biased to improve recognition performance. More recently, [39] has proposed a Bayesian model for top-down visual search and recognition. While these criteria have a solid theoretical justification, the mapping between the saliency algorithms and the neurophysiology of the visual system is usually unclear. Saliency detectors are also typically presented as preprocessing units, strictly tied to low-
level features, whose role is to identify regions to be processed by later recognition stages. This ignores the possibility of hierarchical attention networks, built with various stages of complex features, and the possibility that saliency could play a central role at all stages of recognition. The objective of this work is to explore this possibility.

This is accomplished by the introduction of stand-alone discriminant saliency networks. These are hierarchical networks, where each layer implements a saliency detector based on features of increasing selectivity and invariance. Unlike previous saliency models, these networks account for both preattentive saliency based on simple features, and attentive saliency implemented with more complex features. Each saliency detector is derived from the discriminant saliency principle [56], giving each network layer a clear functional justification. Discriminant saliency also provides a precise optimality criterion for this task, the minimization of recognition error, and an optimal solution, the top-down saliency measure of [57]. Each of the proposed network layers implements this measure with the biologically plausible computations of [56]. This leads to a precise sequence of operations, which is consistent with the standard neurophysiological model [20], and gives the model a computational justification, assigning a precise statistical meaning to all intermediate operations. All parameters can thus be learned optimally with standard statistical procedures, enabling the explicit optimization of the network for recognition.

The performance of the proposed discriminant saliency network architecture is evaluated in three ways. The first is a comparison to other saliency models on an object localization task, where it is shown to achieve the best top-down saliency results reported to date. Along with standard dataset, we consider amorphous object localization where the absence of low level feature compared to the surrounding scenes is distinctive. Discriminant saliency network equating the absence of feature as discriminancy showed significant improvement compared to state-of-arts methods. The second is a comparison to the previous biologically
inspired recognition models on an object recognition task. It is shown that the addi-
tion of a clear optimality criterion for network optimization, as well as the clear
functional justification provided by saliency, lead to substantial improvements over
the best results in the literature. The third is a comparison to the the state-of-
the-art recognition algorithms in the broader computer vision literature. Here, it is
shown that several popular approaches in the literature can actually be mapped
to a network form that resembles the one now proposed. This enables direct com-
parison by setting of comparable parameters the same. The results suggest that
computer vision models achieve gains simply relying on extensions of computing
units which is not necessarily related to model itself. With the comparable com-
puting units discriminant saliency network architecture showed better performance
than state-of-the-art algorithms.

IV.B Multi-layer discriminant saliency networks

In this work, we study the family of Hierarchical discriminant saliency
networks (HDSN). These are members of the the broader family of hierarchical
recognition networks, and generalize the saliency detectors of [56, 57, 65].

IV.B.1 Hierarchical recognition networks

Although HDSNs can have any number of layers, we restrict our atten-
tion to the two-layer hierarchical networks of Figure Figure IV.1. This is mostly
for compatibility with the existing recognition literature, both in biological and
computer vision. Layer $m$ contains a sub-layer $S_m$ of $S$ units and a sub-layer $C_m$
of $C$ units. Each sub-layer is repeated at $R$ image resolutions, as symbolized by
chopped pyramids in the figure. $S$ layers are convolutional, and organized into $C$
channels. Channel $c$ of $S_m$ is based on the dot-product of its input with a template,
$T_c^{(m)}$, shared by all channel units. This is well known to replicate the convolution
of the layer’s input with a filter, which is a transposed replica of $T_c^{(m)}$ [62]. For this
Figure IV.1 Hierarchical feedforward recognition architecture, with two layers and four orientation channels.
reason, we use the terms “filter” and “template” interchangeably. The network of
the figure has \( C_1 = 4 \) channels in layer 1 and \( C_2 = N \) in layer 2. \( S_1 \) units use band-
pass filters \( T^{(1)}_c \), organized into channels \( c \) by filter orientation. The templates of
\( S_2 \) units have the more complex structure \( T^{(2)}_c = \{ T^{(2)}_1, \ldots, T^{(2)}_{C_1} \} \), spanning the
\( C_1 \) channels of the first layer. \( C \) units pool \( S \) unit responses within each feature
channel and scale, over a pooling area \( N \) of simple unit responses, and with some
degree of subsampling \( S \). The pooling neighborhood is usually smaller than the
output of the afferent \( S \) sub-layer. Hence, both \( S \) and \( C \)-units have limited spatial
support. The location \( l \) at the center of the unit’s spatial support is denoted as the
unit’s location. \( C \)-unit pooling can be location adaptive, i.e. \( N(l) \) can depend on \( l \).
The number of scales and channels of \( C \) units is identical to that of \( S \) units. Finally,
the visual stimulus is represented by a vector of \( C_2 \) responses. This hierarchy is
frequently referred to as the “standard model” for object recognition in the visual
cortex, namely the circuitry of the ventral visual pathway for immediate object
recognition [138]. The topology of the network is characterized by the parameters
\( \Theta_m = \{ R_m, C_m, S_m, N_m \} \) for each of its layers.

IV.B.2 HDSN architecture

A HDSN is a hierarchical recognition network whose layers are imple-
mented with the DSN of Section III.C. \( y^{(0)} \) is the network input, and \( y^{(m-1)}_c \) the
output of \( c^{th} \) channel of Layer \( m-1 \). At Layer \( m \), \( y^{(m-1)} \) is first contrast normalized
\[
\overline{y}_c(l) = \frac{y^{(m-1)}_c(l)}{\sum_{j \in Z(l)} \sum_i y^{(m-1)}_i(j)}
\]
where \( Z(l) \) is a window, centered at \( l \), with the size of template \( T^{(m)}_c \). The normal-
ized layer input is then correlated with a set of \( S \)-unit templates \( T^{(m)}_c \), to produce
the feature responses
\[
x^{(m)}_c(l) = \sum_i \left( T^{(m)}_{c,i} \right)^T \overline{y}_i|Z(l).
\]
where \( \overline{y}_i|Z(l) \) is the restriction of \( \overline{y}_i \) to window \( Z(l) \). Note that \( c \) now
refers to the channels of layer \( m \). The template components \( T^{(m)}_{c,i} \) are bandpass
filters of small spatial support. Across these components, the template $T_c^{(m)}$ has unit norm.

S-units are organized by template, as shown in Figure IV.1. Units associated with $T_c^{(m)}$ compute the target posterior map

$$P_c^{(m)}(l) = P_{Y|X_c^{(m)}}(1|x_c^{(m)}(l)) = \sigma[g(x_c^{(m)}(l))] \quad (IV.3)$$

using (III.19) and (V.5). C-units then use (III.7) to compute the saliency maps

$$y_c^{(m)}(l) = S_c^{(m)}(l) = \frac{1}{|N(l)|} \sum_{l \in N(l)} \xi[P_{Y|X_c^{(m)}}(1|x_c^{(m)}(l))], \quad (IV.4)$$

at the output of layer $m$. The $c^{th}$ channel is the saliency map with respect to template $T_c^{(m)}$.

### IV.B.3 Visual representation

In addition to input and resolution, the computations of the two network layers differ in 1) S-unit templates, and 2) C-unit pooling neighborhoods. Since layer 1 templates are simple features, e.g. Gabor filters [138, 117], this layer implements a detector of salient simple features. This is unlike layer 2, which aims to identify, among the patterns of saliency produced by layer 1, those distinctive of the target. This requires complex features, tuned for the detection of the target class, which must be learned. Hence, layer 2 is a detector of discriminant saliency configurations. The importance of capturing relative spatial structure between features within the target is well documented in both the biological and non-biological recognition literatures [117, 93, 4, 147]. Overall, while layer 1 processes visual appearance, e.g. edges, layer 2 captures shape information, e.g. edge configurations.

The different complexity of the templates used by the two layers warrant different C-unit pooling neighborhoods. Since simple features are homogenous, layer 1 relies on a fixed neighborhood $N^{(1)}$, in our implementation a $5 \times 5$ window. On the other hand, to accommodate the diversity of its complex features, layer 2 uses template specific pooling neighborhoods $N_c^{(2)}$, which vary across channels $c^1$.

\footnote{Note that the same neighborhood is used across locations of the same channel.}
These neighborhoods are constrained to a sub-region of their counterparts in layer 1, i.e. $\mathcal{N}_c^{(2)} \subset \mathcal{N}_c^{(1)}, \forall c$, so as to guarantee hierarchical consistency of the network. As discussed in Section IV.B.1, all processing is repeated across multiple image scales. Pooling neighborhoods are adjusted according to scale as illustrated in Figure IV.1, where they are represented by small 3D pyramid structures of fixed size in layer 1, and variable size in layer 2. In all experiments, we use a network with 10 scales, where each scale is $2^{1/4}$ smaller than the previous one. Layer 1 has pooling neighborhoods of depth 2, i.e. $\mathcal{N}^{(1)}$ has size $5 \times 5 \times 2$, and sampling interval $3 \times 3 \times 1$. At layer 2, pooling neighborhoods have depth $d$, where $d$ is randomized from $\{3, 5, 7\}$, and only one location is sampled. This is the location from which the template was extracted during training. In summary, given an $M \times N$ image, the network input is a pyramid of size $M \times N \times 10$. Each layer 1 output channel has size $M/3 \times N/3 \times 9$, while a single value is computed per layer 2 channel.

### IV.B.4 Network training

Network training involves learning the templates $T_c^{(m)}$ of each network layer $m$, and the GGD scale parameters $\alpha_c^{(m)}, \alpha_{c,1}^{(m)}$ of their response. Given $T_c^{(m)}$, the network is exposed to images from class $i \in \{0, 1\}$, and the training samples $\mathcal{R}_{c,i}^{(m)}$ are collected. These consist of the responses $x_c^{(m)}(l)$ of (IV.2) across all locations $l$ and training images in class $i$. The scale parameters are then computed with (III.2). Many approaches are possible to learn the templates $T_c^{(2)}$. We adopt the procedure used to train the second layer of the HMAX network \cite{138, 117}. A number of saliency templates are randomly sampled from $C_1$ layer responses, during training. This consists of extracting patches, centered at random locations and scales, of the $C_1$ response to random target images. Each patch has dimension $n \times n \times 4$, for $n \in \{4, 8, 12, 16\}$, and is normalized to zero mean and unit norm (over the 4 channels). The pooling neighborhood used by $C_2$ units is randomized between 20% and 100% of the size of $s2$. 
IV.C  Relationships to previous models

In this section we study the relationships between the HDSN and previous models in the saliency, biologically inspired vision, and computer vision literatures.

IV.C.1 Saliency models

Most saliency models are stimulus driven, i.e. models of bottom-up saliency. They implement center-surround operations [78, 56], frequency analysis [35], or detect locations of stimuli with specific properties, such as low-probability [180, 18], high entropy [83], or high complexity [137]. Besides the stimulus, it is well known that saliency is influenced by the task to be performed. For example, knowledge of target features increases the efficiency of visual search for a target among distractors [171, 154]. This is known as top-down saliency, and has been classically modeled by modulating features responses [151, 172, 34, 118], i.e. global feature selection. More recent models estimate distributions of feature response to target and background, and use them to derive optimal decision rules, in either the Bayesian [39] or information theoretic sense [57, 65]. These rules modulate feature responses spatially, according to the visual stimulus at different locations.

Information theoretic optimality leads to the saliency measures of (III.13) and (III.7), first used to design a top-down saliency detector in [57]. This detector is a single-layer DSN. More recently, the Bayesian view has been used to derive an alternative model, with two simplifications: 1) assumption of Gaussian instead of generalized Gaussian responses ($\beta = 2$), and 2) use of the target log likelihood $P_{X^{(1)}|Y}(x^{(1)}_c(t)|1)$, instead of the KL divergence of (III.13), as saliency criteria [39]. In network form, this corresponds to three modifications to the DSN: elimination of 1) C units, 2) the sigmoid $\sigma(x)$, and 3) the top divisive normalization branch (see Figure IV.2 a)) of S units. In summary, these models are equivalent to either a single-layer DSN or a simplified version without C units or background
Figure IV.2 DSN computations. a) S-unit : the LLR is computed by divisively normalizing a feature response $x$ differentially, using the outputs of two units that estimate GGD parameters, under the target and null hypothesis. The output non-linearity $\sigma(\cdot)$ transforms the LLR into a posterior target probability. b) C-unit cell computations: responses of afferent simple units are pooled within some region $\mathcal{R}$, after passing them through a non-linearity.

modeling in S units. We refer to these S units as target likelihood (TL) units, and the resulting network as likelihood saliency network (LSN).

IV.C.2 HDSN vs. saliency models

The HDSN implements a top-down saliency principle, which is tuned for discrimination between target and background. Hence, even a single layer DSN outperforms stimulus driven saliency models in object recognition [65] or localization [57] tasks. With respect to top-down saliency, classical feature tuning methods are hampered by the lack of selectivity of the underlying features. These are typically Gabor-like filters which are not selective enough for fine target discrimination. The problem is compounded the inability to localize potential target locations. Due to their selectivity, even globally discriminant features tend to respond to stimuli across the visual field. Finally, they rely on heuristic feature selection criteria, based on “signal-to-noise ratio” measures that do not equate saliency to optimal object recognition or localization.
The HDSN has connections with various biologically inspired recognition architectures. In what follows we consider convolutional networks and HMAX.
**Convolutional neural networks**

The hierarchical network structure of Figure IV.1 has been used since Fukushima’s neocognitron [52]. Early networks suffered from the lack of explicit optimality criterion for training. Convolutional networks became popular in the 1980s [94]. They are a simplified version of the HDSN, with no C units and S units composed uniquely of filtering and the sigmoidal non-linearity of (V.5). Although quite effective for tasks such as digit recognition [94, 110], they have much weaker performance on natural image classification. They also require *global learning* procedures, such as the backpropagation algorithm [94], which are intensive, prone to local minima, and ineffective for deep networks.

Recent extensions introduce S and C-like units in each network layer [123, 80]. S-units include filtering, rectification, and contrast normalization, C-units pool their responses. These extensions have significantly improved performance, sometimes producing staggering improvements. For example, [80] reports that simply rectifying the output of each convolutional network unit drastically improves recognition accuracy. In fact, a network with random filters, but whose S-units include rectification and normalization, performs close to a network with extensively optimized filters. In the network of [80], the input of layer \( m \) is first convolved with a set of filters \( T_c^{(m)} \), producing feature responses \( x_c^{(m)} \). These are passed through a squashing non-linearity, absolute value rectification, subtractive, and divisive normalization, according to

\[
\begin{align*}
a_c^{(m)}(l) & = |g_c \tanh x_c^{(m)}(l)| \quad (IV.5) \\
v_c^{(m)}(l) & = a_c^{(m)}(l) - \sum_{j \in \mathcal{M}(l)} w(j)a_c^{(m)}(j) \quad \sum_{j \in \mathcal{M}(l)} w(j) = 1 \quad (IV.6) \\
u_c^{(m)}(l) & = \frac{v_c^{(m)}(l)}{\max \left( \epsilon, \sum_{j \in \mathcal{M}(l)} w(j) \left( v_c^{(m)}(j) \right)^2 \right)}, \quad (IV.7)
\end{align*}
\]

where \( \mathcal{M}(l) \) is a 9 × 9 window. The normalized responses are finally subject to
spatial pooling

\[ y_c^{(m)}(l) = \sum_{j \in \mathcal{N}(l)} u_c^{(m)}(j) \]  

and subsampled. It is shown that unsupervised learning of the filters \( T_c^{(m)} \) is marginally better than adopting a random filter set, and relatively small gains result from global filter learning.

**The HMAX network**

HMAX networks also belong to the family of Figure IV.1 [138]. \( S_1 \) units are simple Gabor filters, whose responses are pooled by \( C_1 \) units, using the maximum operator

\[ y_c^{(1)}(l) = \max_{j \in \mathcal{N}(1)(l)} x_c^{(1)}(j), \]  

where we again denote responses by \( x_c^{(m)}(l) \) and pooling window by \( \mathcal{N}^{(m)} \). The \( S_2 \) sub-layer is a radial basis function (RBF) network with outputs

\[ s_c^{(2)}(l) = \exp \left( -\beta \| y^{(1)}(l) - T_c^{(2)} \|^2 \right), \quad \| y^{(1)}(l) - T_c^{(2)} \|^2 = \sum_i \| y_i^{(1)}(l) - T_c^{(2)} \|^2 \]  

where \( \beta \) determines the sharpness of the RBF-unit tuning and \( T_c^{(2)} \) is a template. As in the HDSN, these templates are randomly selected during training, and have as many components \( T_{c,i}^{(2)} \) as the number of layer 1 channels. \( C_2 \) units are again max-pooling operators

\[ y_c^{(2)}(l) = \max_{j \in \mathcal{M}(2)(l)} s_c^{(2)}(j), \]  

where \( \mathcal{M}(2)(l) \) is the whole visual field.

**IV.C.4 HDSN vs. biological models**

Despite some similarities with the HDSN, the layers of these networks lack a principled statistical interpretation (e.g. computation of target probabilities) and a functional justification (e.g. saliency detection). Although their templates can be optimized, there is no explicit optimality criteria for the sequence of computations.
In the case of HMAX, some of the network stages are just too simplistic for strong recognition performance. A number of improvements have been proposed in [117], including a lateral inhibition that emulates the effects of divisive normalization and the restriction of $\mathcal{M}^{(2)}(l)$ to a template-specific neighborhood of $l$, to increase the localization of $C_2$ units. Finally, a single set of templates is shared by all object classes, and a support vector-machine (SVM)-based feature selection mechanism used to select the most discriminant subset. Although all these additions have been to shown to enhance performance, they do not address the lack of a clear optimality criterion for the network as a whole. Although the HDSN shares with HMAX the randomized selection of $S_2$ templates, it also implements a number of operations (divisive normalization, non-linearities, etc.) required for decision-theoretical optimality.

Similar remarks can be made for the convolutional network. Consider Figure IV.4, where an HDSN layer is compared to a convolutional network layer [80] (for simplicity, the spatial pooling at the output of both layers is omitted). The operations of the two networks are qualitatively similar (convolutions, contrast normalization, rectification) but performed in different order. While the HDSN sequence is optimal in a well defined sense, this is not the case for the convolutional network. The details of the computations are also quite different. Compare, for example, the rectification operations of b). While [80] uses half-wave rectification, the saliency measure of (III.13) relies on a rectifier that combines (III.7) with (III.17) and (III.16),

$$\Psi(x) = \xi\{\sigma[\gamma|x|^{\beta} - T]\}, \quad \gamma = \left(\frac{1}{\alpha_0^{\beta}} - \frac{1}{\alpha_1^{\beta}}\right), \quad T = \log \frac{\alpha_1}{\alpha_0}.$$  

Note that $\Psi(x)$ depends strongly on the scales, $\alpha_1$ and $\alpha_0$, of feature response under target and background hypotheses, respectively. When $\alpha_1 = \alpha_0$, target and null distributions are identical, i.e. there is nothing to discriminate, and saliency is null for all $x$. Hence, features totally non-informative for target detection are completely inhibited. When $\alpha_1 > \alpha_0$, the target distribution has a heavier tail than the null distribution, i.e. the feature is present in the target. In this case, the
Figure IV.4 Operation sequences per layer of the convolutional network (top) and the HDSN (bottom). The most significant difference is the replacement of half-wave rectification by a saliency-based rectifier, tuned per feature.

The rectifier enhances strong responses and inhibits small ones, acting as a detector of feature presence. Conversely, the null hypothesis has heavier tail when $\alpha_1 < \alpha_0$, i.e., when the feature is absent from the target. In this case, the rectifier enhances small responses and inhibits large ones, acting as a detector of feature absence.

In summary, the rectifier $\Psi(x)$ adapts to the feature distributions under the two hypotheses. This follows from the top-down nature of the saliency computation, and is not possible for the half-wave rectifier of the convolutional network.

This adaptation is reminiscent of optimal rules for image denoising [24]. Like these rules, it thresholds the feature response, exhibiting a dead-zone (region of zero output) which depends on the feature type. This dead-zone reflects the Bayes decision rule of (III.7), for classification of the response $x(l)$ into target and background. Hence, the rectifier is an optimal feature denoising operator for the detection of targets embedded in clutter. The dead-zone depends the relative scales of the target and background distributions, according to

\[
|x|^\beta \leq T/\gamma \quad \text{when } \alpha_1 > \alpha_0 \\
|x|^\beta \geq T/\gamma \quad \text{when } \alpha_1 < \alpha_0.
\]  

(IV.12)
Overall, the HDSN can be seen as a variation of the neocognitron with a precise statistical justification, a well defined optimality principle, a simple optimal training algorithm, a one-to-one mapping to the standard neurophysiological model, and a functional connection to the saliency (and attention) mechanisms of mammalian visual systems. These characteristics can make a significant difference in recognition performance, when compared to the previous networks.

**IV.C.5 popular non-biological models**

Finally, we show that many popular recognition approaches in computer vision are instantiations of a *canonical architecture* closely related to the HDSN.

**Canonical architecture**

While this architecture is identical to the two-stage network of Figure IV.1, the two stages are usually represented in the form of Figure IV.5. The first stage transforms an image into a collection of *descriptors*, usually referred as a *bag-of-features*. These descriptors, denoted $y^{(1)}(l)$, can be calculated at many image locations $l$, e.g. per pixel, in a regular pixel grid (dense sampling), or at keypoint locations [105]. We assume dense sampling, because it appears to produce the best results [178], and keypoint selection reduces to inhibiting some network units. Descriptors are high-dimensional vectors, obtained by application of spatially localized operators at each image location. If each descriptor dimension $y_c^{(1)}$ is used to define a channel of this representation, *descriptor channels* are equivalent to the channels of $C_1$ output in Figure IV.1.

The second stage computes statistics of descriptor *assignments* to a set of *descriptor templates*, $T_c^{(2)}$ learned from a training dataset. These templates provide an approximation to the *descriptor probability distribution*, e.g. a mixture model, kernel-density estimator, vector quantizer, or RBF network [38]. While this approximation determines the template learning procedure and assignment rule, the latter is usually based on the *posterior probability* of the descriptor under
Figure IV.5  Canonical architecture implemented by various popular object recognition methods.

the model component defined by each template. This boils down to a measure of descriptor-template similarity $s(y^{(1)}(l), T^{(2)}_c)$ and post-processing. The map of descriptor assignments to template $T^{(2)}_c$ is the $c^{th}$ channel of the representation. Assignment channels are finally pooled into an assignment histogram. The pooling can be over the entire channel, sub-areas, or both.

Stage 1, descriptors

Popular descriptors, e.g. SIFT or HoG, measure image energy along different orientations. We discuss SIFT in detail, but a similar analysis applies to other such descriptors. The SIFT descriptor $y \in R^{128}$ consists of 8-bin histograms of orientation response, computed over $4 \times 4$ pixel cells. The orientations are estimated from image intensity gradients. Location $l$ contributes to histogram bin $k$ with $a_k(l) = r(l)g(l)b_k[\theta(l)]$, where $r(l), \theta(l)$ are the magnitude and orientation components of the gradient at $l$, $g(l)$ a Gaussian weight that penalizes locations farther from the descriptor center, and $b_k(\theta)$ a trilinear interpolator, based on the
distance between $\theta$ and the central orientation of the $k^{th}$ bin. The $k^{th}$ histogram entry is

$$h_k = \sum_{l \in B} a_k(l),$$

where $B$ is the $4 \times 4$ pixel cell. The descriptor concatenates histograms of $4 \times 4$ cells into a 128-dimensional vector, which is normalized, fed to a saturating nonlinerity $\tau(x) = \max(x, 0.2)$ and normalized again to unit length. Using superscripts $q$ to denote cells, and subscripts $k$ to denote orientation bins, this is equivalent to the sequence of computations

$$\overline{h}_{qk} = \tau \left[ \frac{h_{qk}^q}{\sum_{q,k} h_{qk}^q} \right] = \tau \left[ \frac{\sum_{l \in C_q} a_k(l)}{\sum_{q,k} \sum_{l \in C_q} a_k(l)} \right]$$

$$s_{qk} = \frac{h_{qk}^q}{\sum_{q,k} h_{qk}^q} \quad k \in \{1, \ldots, 8\}, q \in \{1, \ldots, 16\}$$

$$y = (s^1, \ldots, s^{16})^T.$$

Since (IV.15) combines divisive normalization of $a_k(l)$ by all responses in the $4 \times 4$ cells, average pooling, and squashing non-linearity, it can be mapped to a sequence of S and C unit operations similar to (IV.7) and (IV.8). The main difference is the application the non-linearity after pooling vs. after filtering, as in (IV.5). Note that (IV.15) can be seen as stage 2 pre-processing, which contrast normalizes stage 1 responses. This is identical to (IV.1), the normalization of HDSN layer inputs.

**Stage 2, descriptor assignments**

The descriptor templates of stage 2 can be learned in a number of ways. For example, MPE [161] and HGMM [181] use mixture models, SPMK [93] clustering techniques, such as k-means [85, 38, 67], sparse SPMK [174] sparse dictionary learning [90], and NBNN stores every training descriptor. Although this leads to different terminology for descriptor templates, including *visual words*, [30, 142], *textons* [96], *visemes* [42], these methods are all closely related. Most methods...
assign descriptor $y^{(1)}(l)$ to template $T_c^{(2)}$, according to probabilities evaluated with a Gaussian mixture model of means $T_c^{(2)}$, $c \in \{1, \ldots, N\}$

$$P_{Y^{(1)}}(y^{(1)}(l); \beta, T^{(2)}) = \frac{1}{N} \sum_{c=1}^{N} \exp \left( -\beta \| y^{(1)}(l) - T_c^{(2)} \|^2 \right) = \frac{1}{N} \sum_{c=1}^{N} s_c^{(2)}(l),$$

(IV.17)

where $s_c^{(2)}(l)$ are the HMAX likelihoods, of (IV.10). Hence, the descriptor likelihoods per Gaussian component of HMAX are replaced by posterior probabilities of component given descriptor

$$p_c^{(2)}(l) = \frac{s_c^{(2)}(l)}{\sum_{c=1}^{N} s_c^{(2)}(l)}.$$

Defining the template responses

$$x_c^{(2)} = (T_c^{(2)})^T y^{(1)}(l),$$

(IV.18)

this can be written as

$$p_c^{(2)}(l) = \frac{\exp \left( 2\beta x_c^{(2)}(l) - \beta \| T_c^{(2)} \|^2 \right)}{\sum_{c=1}^{N} \exp \left( 2\beta x_c^{(2)}(l) - \beta \| T_c^{(2)} \|^2 \right)}$$

If the templates have unit norm, $\| T_i^{(2)} \| = 1$, $\forall i$, it is equal to

$$p_c^{(2)}(l) = \mu_{-\beta}(x_c^{(2)}(l)) \quad \mu_{\beta}(x_c) = \frac{\exp(-2\beta x_c)}{\sum_{j=1}^{N} \exp(-2\beta x_j)},$$

(IV.19)

where $\mu_{\beta}(x)$ is the softmax function. Hence, $p_c^{(2)}(l)$ is the response of a softmax network with weights defined by template $T_c^{(2)}$ to descriptor $y^{(1)}(l)$. Some methods, e.g. MPE, HGMM, or NN, learn descriptor templates per object class and compute the posterior class probability

$$P_{Y|X}(c|y^{(1)}(l)) = \sum_{j \in \mathcal{I}_c} p_j^{(2)}(l)$$

(IV.20)

where $\mathcal{I}_c$ is the set of indices of templates from class $c$.

In summary, the various approaches are equivalent to combinations of the RBF network of HMAX and different forms of output normalization. This leads to
Table IV.1  Comparison of comparable units in various architectures with HDSN: first layer

<table>
<thead>
<tr>
<th>Method</th>
<th>Stage 1</th>
<th>units</th>
<th>saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDSN</td>
<td>DS</td>
<td>top-down</td>
<td>-</td>
</tr>
<tr>
<td>HMAX [138]</td>
<td>filter responses</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>MPE [22]</td>
<td>filter responses</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>NBNN [12]</td>
<td>SIFT</td>
<td>bottom-up</td>
<td></td>
</tr>
<tr>
<td>SPMK [93]</td>
<td>SIFT</td>
<td>bottom-up</td>
<td></td>
</tr>
<tr>
<td>HGMM [181]</td>
<td>SIFT</td>
<td>bottom-up</td>
<td></td>
</tr>
<tr>
<td>sparse SPMK</td>
<td>SIFT</td>
<td>bottom-up</td>
<td></td>
</tr>
<tr>
<td>DSN [57]</td>
<td>DS</td>
<td>top-down</td>
<td></td>
</tr>
<tr>
<td>LSN [39]</td>
<td>SL</td>
<td>top-down</td>
<td></td>
</tr>
</tbody>
</table>

Table IV.2  Comparison of comparable units in various architectures with HDSN: second layer

<table>
<thead>
<tr>
<th>Method</th>
<th>Stage 2</th>
<th>templates</th>
<th>units</th>
<th>assignment</th>
<th>pooling</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDSN</td>
<td>random</td>
<td>complex</td>
<td>DS</td>
<td>-</td>
<td>sum</td>
</tr>
<tr>
<td>HMAX</td>
<td>random</td>
<td>simple</td>
<td>LU</td>
<td>soft</td>
<td>max</td>
</tr>
<tr>
<td>MPE</td>
<td>GMM</td>
<td>complex</td>
<td>CPU</td>
<td>soft</td>
<td>sum</td>
</tr>
<tr>
<td>NBNN</td>
<td>training set</td>
<td>simple</td>
<td>CPU</td>
<td>hard</td>
<td>sum</td>
</tr>
<tr>
<td>SPMK</td>
<td>codebook</td>
<td>simple</td>
<td>PU</td>
<td>hard</td>
<td>sum</td>
</tr>
<tr>
<td>HGMM</td>
<td>GMM</td>
<td>complex</td>
<td>CPU</td>
<td>soft</td>
<td>sum</td>
</tr>
<tr>
<td>sparse SPMK</td>
<td>sparse dictionary</td>
<td>simple</td>
<td>PP</td>
<td>soft</td>
<td>max</td>
</tr>
<tr>
<td>DSN</td>
<td></td>
<td></td>
<td>DS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSN</td>
<td></td>
<td></td>
<td>SL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

three types of layer 2 units: the likelihood units (LU) $s^{(2)}_c(l)$ of (IV.10), the posterior units (PU) of (IV.19) or the class-posterior units (CPU) of (IV.20). The precision parameter $\beta$, controls the softness of the assignments. When $\beta \to 0$ the mixture model of (IV.17) becomes a vector quantizer [163] and $p^{(2)}_c(l) = 1$ for the template closest to $y^{(1)}(l)$, and zero for all others. This is denoted a hard-assignment. When $\beta > 0$ descriptors are assigned to multiple representatives, according to the posteriors $p^{(2)}_c(l)$. This is a soft assignment. Table IV.1 and Table IV.2 summarizes the architecture of the different networks. Note that the method used to learn representatives is not determinant. For example, HMAX templates could
be learned with k-means, and SPMK codebooks built with randomly collected examples. In fact, a variety of methods have been proposed for codebook learning alone [30, 44, 115, 142, 170].

The main exception to this rule is sparse SPMK [174], which directly computes the assignments \( p_c^{(2)}(l) \) by solving the optimization

\[
p^{(2)}(l) = \arg \min_p \|y^{(1)}(l) - T^{(2)} p\|^2 + \lambda \|p\|_1 \quad \text{(IV.21)}
\]

where \( T^{(2)} \) is a dictionary with templates \( T_c^{(2)} \) as columns, \( \|p\|_1 \) the \( \ell_1 \) norm of \( p \), and \( \lambda \) a regularization parameter. This produces a soft assignment, of sparsity (number of non-zero entries) controlled by \( \lambda \). \( T^{(2)} \) is learned by minimizing this cost over all \( y^{(1)}(l) \), alternating between the minimization with respect to \( T^{(2)} \) and with respect to all \( p(l) \). An extension that learns discriminant dictionaries is presented in [79]. While there is some evidence that sparse assignments are effective for recognition, the price is a substantial increase in computation, since the optimization of (IV.21) has to be repeated for each descriptor of the image to classify. This is frequently done with greedy optimization by matching pursuits [108] which involve multiple iterations over all templates in \( T^{(2)} \). We denote SPMK units as projection pursuit (PP) units.

The final operation of stage 2 is to derive, from the assignment maps, a measure of confidence on the presence of each template in the image. This is usually the average probability

\[
y_c^{(2)}(l) = \frac{1}{|N^{(2)}(l)|} \sum_{m \in N^{(2)}(l)} p_{2}^{(2)}(m), \quad \text{(IV.22)}
\]

although some methods use the maximum operator of (IV.11). The neighborhood \( N^{(2)}(l) \) can be the entire image, in which case there are as many pooling units as representative descriptors, i.e. \( N \), but is usually repeated for a number of subregions, using the pyramid structure introduced by SPMK. The most common configuration is a three-layer pyramid, containing the full image at level 0, and its partition into \( 2 \times 2 \), and \( 4 \times 4 \) equal sized cells at levels 1 and 2, respectively. In this case, there are a total of \( 21N \) pooling units.
**Statistical interpretation**

A statistical interpretation of the canonical architecture of Figure IV.5 is possible if, in the SIFT descriptor, \(a_k(l)\) is replaced by the response magnitude \(|x_k^{(1)}(l)|\) of a Gabor filter with the orientation of the \(k^{th}\) histogram bin. Note that the two quantities are conceptually equivalent: both measure image energy along orientation \(k\). The use of \(|x_k^{(1)}(l)|\), with the machinery of Section III.C, enables a statistical interpretation of the computations. Defining

\[
\alpha = \sum_{q,j} \sum_{l \in C_q} |x_j^{(1)}(l)|
\]

\[
= \sum_{q,l \in C_q} |x_k^{(1)}(l)| \geq \sum_{j \neq k \sum_{q,l \in C_q}} |x_j^{(1)}(l)|
\]

\[
= \sum_{q,l \in C_q} |x_k^{(1)}(l)| + \nu
\]

the SIFT computation reduces to

\[
\bar{h}_k^q = \sum_{l \in C_q} |x_k^{(1)}(l)| / \alpha
\]

\[
\propto - \sum_{l \in C_q} \log P_{X_k^{(1)}}(x_k^{(1)}(l); \alpha, 1)
\]

\[
\propto - \int_{C_q} P_{X_k}(x; \alpha_q, 1) \log P_{X_k}(x; \alpha, 1) dx
\]

with \(P_{X}(x; \alpha, 1)\) as given in (III.1), and \(\alpha_q = \sum_{l \in C_q} |x_k^{(1)}(l)|\). Hence, up to constants, \(\bar{h}_k^q\) measures the cross-entropy between the responses of filter \(X_k^{(1)}\) within cell \(C_q\) and across the support of the descriptor. Assuming that the distributions are identical, this is the response entropy.

Interestingly, entropy is a common saliency measure [130, 180, 18], which equates salient to rare (low-probability) events in the visual field. The SIFT descriptor can thus be seen as a *saliency measure*, identifying as salient the locations of stimuli whose orientation is rare within a local neighborhood. This is a *strictly bottom-up* definition of saliency. The descriptor templates \(T_c^{(2)}\) are *saliency templates* and, by assigning descriptors to templates, stage 2 groups image locations.
by similarity of *saliency patterns*. The assignment histogram at the network output estimates the relative probabilities of these patterns in the image to classify.

IV.C.6  **HDSN vs. computer vision models**

The HDSN is naturally mapped into the canonical architecture of Figure IV.5 by equating each of its layers to one architecture stage. At stage 1, descriptor channels are the channels of $C_1$ unit output. While these are saliency maps, they identify feature responses *discriminant* for target detection. Hence, the advantages of the HDSN over SIFT-based methods include the benefits of top-down over bottom-up saliency: saliency templates that are highly *selective* for the target. When the computations of the units in the two networks are compared, this is compounded by the advantages of the HDSN over the convolutional network: feature denoising by an adaptive rectifier tuned to the discriminant power of each feature, modulation of saliency responses by the discriminant power of the underlying features, and ability to detect both feature presence and absence.

In stage 2, the HDSN assigns descriptors to templates according to the target posteriors $P_{Y|X^{(2)}_c}|X^{(2)}(l)$ of (IV.3). This is similar to the posterior class probabilities (IV.20) of MPE, but produces a probability *per saliency template* $T^{(2)}_c$. The templates are themselves different. While those derived from SIFT tend to depict *simple features*, namely edges of locally dominant orientation, HDSN templates are *complex features* [165, 54], e.g. *configurations* of edges that are discriminant for target detection. Among the non-biological approaches, only MPE achieves an equivalent effect, by learning multiple simple features per class and combining the individual posteriors. The HDSN is a more *efficient* representation, implementable with much fewer layer 2 units. Note that SPMK-style methods have no access to complex features or discriminant templates, in any stage. Finally, there are some differences on the collection of assignment statistics. For each location $l$, HDSN pooling is determined by the neighborhood $\mathcal{N}^{(2)}(l)$, according to (III.7). On the other hand, non-biological methods compute the average or max posterior
probability over the fixed neighborhoods of SPMK.

IV.D Distinguishing characteristics of the HDSN architecture

In this section, we discuss the advantages of the HDSN over the approaches discussed in the previous sections.

IV.D.1 General differences

We start by noting that, although it can be interpreted as a model of biological recognition, the HDSN is more than just a model. As discussed in Sections III.C and IV.B, each of its simple units computes the posterior probability of target presence at an image location, given the feature responses observed in a neighborhood of that location. This is, up to a threshold, the optimal decision rule for the detection of the target at the location, in the MPE sense [38]. Complex units then measure the discriminant power of each feature, at each location, for target/background classification. The particular measure of discrimination, the KL divergence between distributions of feature response under the target and background hypotheses, is a close approximation to the Bayes error of the target detection problem [57]. Hence, the computations of each network unit are optimal under a well defined detection criterion, and have a well defined functional justification: the network provides the optimal measurements for both target detection and feature selection, at each location of the visual field. In this sense, the HDSN is much more similar to previous non-biological classifiers than to previous biologically inspired convolutional networks or saliency models. In what follows, we analyze these issues in greater detail.
Figure IV.6 half-wave (left) and saliency-based (right) rectifiers. The latter varies significantly with the relative values of $\alpha_1$ and $\alpha_0$.

**IV.D.2 Characteristics of discriminant saliency**

The characteristics of discriminant saliency becomes clear when it is compared with saliency equated as detection of low probability features. Rosenholts [130], Zhang et al. [180], and Bruce and Tsotsos [18] used detection of features of low probability as a criterion for visual saliency. This is also closely related to the most popular strategy for the detection of interest points. As one of the example, entropy at the location $l$ averaged over the visual field $R_l$ is

$$H(l) = -\int_{x \in R_l} P_X(x) \log P_X(x) dx$$

$$(\text{IV.24})$$

$$= K + \frac{1}{2N_{R_l}} \sum_{x_i \in R_l} \left(\frac{|x_i|}{\alpha}\right)^\beta$$

Compared with entropy based saliency measure, $H(l)$, discriminant saliency, $S(l)$, has quite dynamic characteristics depending on the relative value between $\alpha_1$ and $\alpha_0$. Basically, the characteristic function has two types depending on whether $\alpha_1 > \alpha_0$ or vice versa. When $\alpha_1 = \alpha_0$, two distribution is the same meaning that there is nothing discriminant so saliency becomes zero for all $x$. When $\alpha_1 > \alpha_0$, target distribution has high tail than non-target, saliency is high with high responses, on the other hand, saliency is low with high responses when non-target has higher tail.

Figure IV.6 shows saliency response characteristics with respect
to the one datapoint (filter response), $x$, for different value of $\alpha_0$ when $\alpha_1 = 1$ and $\beta = 0.5$ and entropy when $\alpha = 1$ and $\beta = 0.5$ in equation IV.24. When $\alpha_1 > \alpha_0$, the bigger the differences are the higher saliency value for high response $|x|$ and the more datapoints have non-zero values. When $\alpha_1 < \alpha_0$, on the other hand, similar observation can be made; the bigger the differences, the higher saliencies overall with more non-zero datapoints but in opposite polarity.

where $K = \log 2\alpha \Gamma(1/\beta)/\beta$, and $\alpha$ is the GGD scale for the feature responses across the visual field. Note that this is, in many aspects, similar to the measure of discriminant saliency $S(l)$. It is, in fact, equivalent in the limit of $\alpha_1 \to \infty$, if $\alpha = \alpha_0$ and the non-linearities $\sigma(.)$ and $\xi(.)$ are replaced by the identity map. While this may appear a small difference, it has a major impact on the ability of the saliency detector to switch between the detection of feature presence and absence.

The two behaviors are illustrated in Figure IV.6 a), which presents the characteristic function of the discriminant saliency detector - curve of $S(l)$ as a function of $x(l)$ - for different values of $\alpha_1$, when $\alpha_0 = 1$. Note how 1) the presence of the feature $X$ (large feature response $|x|$) elicits a strong saliency response when $\alpha_1 > 1$, but 2) strong saliency responses are reserved for feature absence (small $|x|$) when $\alpha_1 < 1$. For comparison, the characteristic curve of the entropy-based detector is shown Figure IV.6 b), for $\alpha = 1$. In this case, all degrees of freedom are exhausted once the background distribution is fixed, and the saliency detector is always a detector of feature presence.

Note that $S(l)$ also involves weak classifier itself by only considering sample points classified as target. $g(x)$ in equation V.3 is the log likelihood ratio as in equation III.16, therefore, testing $g(x) \geq 0$ makes it only select point $x$ which has higher probability for target $Y = 1$. The range of $x$ which have non zero value is the function of two parameters $\alpha_0$ and $\alpha_1$.

$KL[P_{X|Y}(x|1)||P_{X|Y}(x|0)]$ and is shown to give best performance among other possible combinations [65]. Comparing filter rectification which is simple
absolute function, saliency map suppresses near zero responses and this cut off value is feature specific and tuned from target vs non-target distribution. Note how saliency responses emphasize the accordion, suppressing filter responses due to the person, which is not in the target class.

**IV.D.3 SIFT and bag-of-features**

In computer vision, various state-of-the-art approaches to recognition are based on some combination of the SIFT descriptor and the BoF representation. This combination can be mapped to a two-layer network, where layer 1 computes descriptors and layer 2 the BoF. In layer 1, descriptors are obtained by a combination of Gabor filtering and histograming of orientation responses. SIFT relies on simple cell channels of 8 orientations, and complex cells that increase invariance to local shifts. Pooling is sum-based, and orientation histograms are computed at the complex cell outputs. This is a form of contrast normalization cross orientation channels similar to the contrast normalization shown in neural network [80] of figure Figure IV.4 top part. Functionally, the SIFT descriptor can be seen as a measure of orientation dominance, since histogram bins of larger amplitude correspond to dominant orientations. The descriptor is computed across 4 × 4 neighborhoods, and captures a spatial pattern of dominant edge orientations. It can thus be interpreted as a saliency template, where saliency is equated (in a strictly bottom-up manner) to orientation dominance. One distinguishing feature of SIFT is that it estimates the scale of the visual stimulus at each location when it is combined with interest point detection rather than dense sampling, and adapts the complex cell pooling to this scale [105]. The BoF layer vector quantizes SIFT descriptors, assigning each descriptor to the closest template within a vector quantizer, usually learned with k-means. This is identical to assigning the descriptor to the RBF unit of largest posterior probability, in a set of RBF units of zero variance centered at the templates. Some variants rely on soft assignment of descriptors to templates, i.e. computing the posterior distribution for a set of RBF units of
non-zero variance. In summary, layer 2 consists of S units that measure distances to saliency templates, and C units that pool these distances over predefined spatial neighborhoods (sum pooling). These computations are similar to those of HMAX.

Second stage for both case is the stage to measure if image has specific local structure. Given the representative visual word usually calculated as the cluster center of 128 dimension of SIFT with kmean from SIFT key points, output of S1 is assigned to one (hard quantization) or many (soft quantization) of the selected visual words. This operation is an extreme case of normalization cross patch(codebook) channels, which HMAX type architecture does not have.

IV.D.4 Nearest-neighbor classification

High recognition accuracy has been recently reported for a nearest neighbor (NN) classifier [12], which can also be implemented as a two-layer network. Images are represented as collections of SIFT descriptors, leading to the first layer of the previous section. The second layer then models the probability density of the target class, and uses it to implement the BDR for target detection. Given
an image to classify, this consists of finding the class $y^*$ that maximizes the log-likelihood $\sum_i \log P_{X|Y}(x_i|y)$, of feature vectors $x_i$ extracted from the image. The densities $P_{X|Y}(x|y)$ are learned with a kernel density estimator, which is approximated by the Gaussian closest to each of the feature vectors $x_i$ to classify. Hence, the classifier can also be interpreted as a NN classifier, where the $x_i$ is assigned to the class of the closest vector in the training set. [12] denotes this operation a NN classification with an image-to-class distance. As in BoF, the NN search can be implemented by an RBF layer with hard assignments (zero variance). The main difference is that, instead of a single vector quantizer learned from all classes, the RBF templates are now class-specific, consisting of all descriptors extracted from all training images for the class. This has some commonality with the second HMAX layer, where templates could also be learned per class. However, 1) the HMAX network only stores a small subset of the training examples (random subsampling), and 2) the contribution of template $P_j$ to the RBF computation is $||x_i - P_j||^2$ under NN (distance $\log P_{X|Y}(x_i|y)$) and $\exp(-\beta||x_i - P_j||^2)$ under HMAX (distance $P_{X|Y}(x_i|y)$). The classification rules are then

$$y^* = \arg\min_y \sum_i \sum_j w_{jy} ||x_i - P_j||^2$$

where $w_{jy} = \begin{cases} 1 & \text{P}_j \text{ closest template from class } y \\ 0 & \text{otherwise} \end{cases}$

for NN, and

$$y^* = \arg\max_y \sum_i \sum_j w_{jy} \exp(-\beta||x_i - P_j||^2)$$

for HMAX, where $w_{jy}$ is learned with a linear SVM. The main limitation of the NN classifier is the large number of RBF templates that must be stored and searched. Recently, [181] proposed an alternative NN classifier, which replaces the kernel density estimator with a hierarchical gaussian mixture model (HGMM) composed of a much smaller number of RBF units. While the HGMM is still estimated per class this has far less complexity than [12]. Images are represented by a vector
that combines the RBF posterior and the parameters of the RBF itself. A further stage of discriminant dimensionality reduction is applied, and a NN classifier implemented in the resulting space.

Since the HGMM has a much smaller number of components than the number of training examples, this has far less complexity than the NN classifier of [12].

HGMM can be viewed as general case of BoF representation. In HGMM, codebook is modeled as gaussian mixture and each individual image is also modeled as gaussian mixture where the prior for the image specific parameters comes from the entire image GMM. Both of the tuned gaussian parameters for each image and local sum of the each mixture probability map is used as super-vector. From this huge dimension of super-vector, discriminant attribute projection is performed to reduce dimensionality. This corresponds to sophisticated second layer complex cell rather than simple sum pooling operation. The main difference of this paper is that the codebook is tuned to each image statistics; each image is again represented by predefined codebook but adapting the variation of individual image. This inspires interesting extention for tuned image patch but we did not pursue this aspect in this paper.

IV.D.5 Unified spatial grid vs sampling location based window

One other interesting approach in BoF representation is spatial pyramid representation where histogram is obtained from each subset of image location divided with unified grid. This method shows balance between “subdividing” and “disordering” [93]. In the network framework this corresponds to the size of pooling window in complex cell. Since one sampled patch is used as “visual word” in DiscSal, the information from original location of sampled patch can be used for determining C2 pooling window. Experiment also shows uniformly divided pooling and sampled location based pooling cases. Table IV.3 shows comparison between general DiscSalNet architecture with SIFT+BoF+spatial pyramid case.
Table IV.3 DiscSalNet vs BoF with SPK case. This table shows comparison between general DiscSalNet architecture with SIFT + BoF + spatial pyramid case.

<table>
<thead>
<tr>
<th></th>
<th>DiscSalNet</th>
<th>SIFT+BoF+spatial pyramid</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>selected band pass filter bank</td>
<td>8 orientation filter bank</td>
</tr>
<tr>
<td>C1</td>
<td>fixed pooling 3D window</td>
<td>adaptable pooling window on selected scale followed by</td>
</tr>
<tr>
<td>S2</td>
<td>filtering with multi channel patch on normalized C1 output</td>
<td>distance computing followed by quantization than saliency representation</td>
</tr>
<tr>
<td>C2</td>
<td>pooling window centered on the original location of the sampled patch</td>
<td>Uniformly divided pooling window</td>
</tr>
</tbody>
</table>

IV.E Experiments

In this section, we present an extensive experimental evaluation of the performance of the HDSN on object recognition and localization tasks.

IV.E.1 Object recognition experiments

We first consider object recognition. Despite the availability, in the literature, of extensive evaluations, it is not easy to assess the relative merits of different architectures. The problem is a total lack of uniformity in parameter settings. When different methods are mapped to network form, it is clear that the standard configurations disagree even in the most elementary parameters, e.g. the number of layer 2 units. For example, SPMK usually relies on a codebook of size 1,024 and a pyramid of 21 pooling regions. While this should be compared to an HMAX network of $21K$ units, only $4K$ are usually adopted in the HMAX literature. NN methods further increase the number of units by one or two orders of magnitude, since a codebook is learned per class and standard datasets have between tens and hundreds of classes. In the worst case of [12] (as many units as training examples), results are presented for an RBF of 10 million units. This lack of uniformity is problematic since, for any method, recognition performance
tends to increase drastically with the number of units. Obviously, comparisons across architectures are meaningless unless this variable is precisely calibrated. This calibration should be extended to all architectural variables, e.g. feature set, SVM used at the network output, etc.

By mapping many approaches into a common architecture, the discussion of the previous sections enables a controlled evaluation. We divided this evaluation into two sets of experiments. The first aimed to measure the impact of the units of Table IV.1 and Table IV.2 on recognition accuracy. Starting from a minimalist two-layer network, the units were gradually introduced, and network performance recorded. The second was designed to compare the overall performance of the HDSN to the best results in the literature. In this case, the different methods used the parameters specified by their authors, and differed along several dimensions (e.g. different features, units, and network sizes). It is assumed that these parameters were optimized so as to achieve the best results per method. This allows an unbiased comparison of the best possible performance per architecture.

All experiments were performed on the popular Caltech101 (C101) [97] and 15 natural scenes (N15) [93] datasets. C101 contains 9,146 images from 101 object classes and was used with the experimental protocol of [117]: multiclass recognition, with 30 training images per class, and a maximum of 50 for test. N15 contains 4,485 images from 15 scene classes, with 200 to 400 images per class, and an experimental protocol detailed in [93]: 100 images per class randomly selected for training, the rest used as test set. In all experiments, recognition rates were averaged over 5 runs on independent train and test sets. A linear SVM was used to classify the vector of network outputs in all cases. This vector was “sphered” before classification, by normalizing the mean and variance of each dimension to zero and one, respectively [117].
IV.E.2 Impact of network units on recognition performance

The first experiment tested the impact of layer 1 units on recognition accuracy. We started with the following base network.

1. local normalization of image intensities, according to (IV.1);

2. \(S_1\) units: \(C_1 = 4\) Gabor filters of orientations \(\theta \in \{0, \pi/4, \pi/2, 3\pi/2\}\), and \(R_1 = 10\) resolutions \(r \in \{2^{i/4} | i = 0, \ldots, 9\}\);

3. \(C_1\) units: average pooling over a neighborhood \(N^{(1)} = 5 \times 5 \times 2\) with subsampling of \(S_1 = 3 \times 3 \times 1\);

4. \(S_2\) units: 40 LUs with randomly selected templates \(T^{(2)}\) per class, for a total of \(C_2 = 600\) channels for N15 and \(C_2 = 4040\) channels for C101.

5. \(C_2\) units: average pooling over a neighborhood \(N^{(2)} = S_c \times S_c \times d\), centered at the location of the template. \(S_c\) randomized from \(\{10\%, 20\%, 30\%\}\) of template size and the depth \(d\) from \(\{3, 5, 7\}\);

In each setting of the experiment, the \(S_1\) and \(C_1\) units were replaced with those on the left of Table IV.4. The same Gabor channels were used across settings, contrast normalization (CN) was implemented with (IV.5)-(IV.8), SIFT with (IV.15)-(IV.15), and discriminant saliency with (IV.2)-(IV.4). The pooling operator was that which performed best for each network. Note that the type II network is identical to HMAX [138], the first layer of type III is identical to that of the convolutional network of [80], and that of type IV is identical to the first stage of [93, 174, 12, 181].

The table supports several conclusions. First, pooling has a significant impact in recognition performance, confirming the importance of the spatial invariance attributed to this operation, and of complex units in general. Second, contrast normalization enables significant recognition gains. These are larger in C101 than in N15, but always present. Among the non-discriminant methods, SIFT units outperform those of convolutional networks. Finally, DS units combine...
Table IV.4  First layer unit comparison. This experiments are done keeping the second layer operation the same and compare results only on the first layer unit.

<table>
<thead>
<tr>
<th>Type</th>
<th>simple unit</th>
<th>pooling</th>
<th>L2</th>
<th>N15</th>
<th>C101</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>filter</td>
<td>-</td>
<td>LU</td>
<td>58.7 ± 0.3</td>
<td>40.5 ± 0.8</td>
</tr>
<tr>
<td>II</td>
<td>filter</td>
<td>max</td>
<td>LU</td>
<td>65.6 ± 1.3</td>
<td>52.8 ± 1.0</td>
</tr>
<tr>
<td>III</td>
<td>CN</td>
<td>max</td>
<td>LU</td>
<td>67.1 ± 1.5</td>
<td>58.8 ± 1.3</td>
</tr>
<tr>
<td>IV</td>
<td>SIFT</td>
<td>average</td>
<td>LU</td>
<td>67.5 ± 0.6</td>
<td>62.8 ± 0.9</td>
</tr>
<tr>
<td>V</td>
<td>DS</td>
<td>average</td>
<td>LU</td>
<td>68.3 ± 0.6</td>
<td>64.2 ± 1.3</td>
</tr>
</tbody>
</table>

Table IV.5  Second layer unit comparison. This experiments are done keeping the first layer operation the same and compare results only on the second layer unit.

<table>
<thead>
<tr>
<th>Type</th>
<th>L1</th>
<th>L2</th>
<th>N15</th>
<th>C101</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>DS</td>
<td>CPU</td>
<td>67.4 ± 1</td>
<td>61 ± 0.8</td>
</tr>
<tr>
<td>II</td>
<td>DS</td>
<td>PU</td>
<td>68.1 ± 1</td>
<td>62.1 ± 1.1</td>
</tr>
<tr>
<td>III</td>
<td>DS</td>
<td>LU</td>
<td>68.3 ± 0.6</td>
<td>64.2 ± 1.3</td>
</tr>
<tr>
<td>IV</td>
<td>DS</td>
<td>DS</td>
<td>80 ± 0.6</td>
<td>69.2 ± 1.3</td>
</tr>
<tr>
<td>V</td>
<td>DS*</td>
<td>DS</td>
<td>82.2 ± 0.7</td>
<td>69.9 ± 1.7</td>
</tr>
</tbody>
</table>

the best overall recognition performance with the lowest complexity. While the networks of type III and IV perform contrast normalization both within (spatially) and across channels, DSN units only require within channel normalization. This enables independent processing of the different channels, considerably simplifying the implementation of this architecture.

To test the impact of layer 2 on recognition performance, we then selected the layer 1 of best performance, i.e. DS units and average pooling. Besides likelihood units (LU), layer 2 was implemented with posterior units (PU), class-posterior units (CPU), and DS units. Since the number of layer 2 channels is drastically reduced when CPU units are used (from the number of templates to the number of classes, e.g. 600 to 15 in N15 and 4040 to 101 in C101), and this reduces the effectiveness of the SVM that follows the network, we tried alternative CPU configurations. Best results were obtained, in preliminary experiments, by weighing PU units according to the posterior class probability, i.e. multiplying (IV.19) by (IV.20). The accuracies of the different units in the second layer are
summarized in the Table IV.5, network types I-IV. Interestingly, neither PU nor CPU improved the performance of LU. This suggests that cross-channel normalization is not beneficial at layer 2. Again, DS units achieved the best performance, with substantially higher recognition accuracy than LUs. An overall comparison of this network (HDSN) with the HMAX network (type II) on Table IV.4, shows a drastic improvement from 66% to 80% accuracy on N15 and from 53% to 69% on C101. This confirms that the choice of units can have a dramatic impact on network performance, and is strong evidence that discriminant saliency substantially strengthens recognition.

IV.E.3 Comparison with state-of-art architectures

We next compared the performance of the HDSN to the best results reported for the other methods. These are obtained with various extensions of the network of the previous section. We start by introducing a similar optimization for the HDSN.

Discriminant simple features

It is well known that top-down feature modulation influences saliency judgements. For example, knowledge of target features increases the efficiency of visual search for a target among distractors [171, 154]. This process has been modelled in three ways. The first is to estimate the distributions of feature response to target and distractors, and use them to derive optimal decision rules, in either the Bayesian [39] or information theoretic sense [57, 65]. These modulate feature responses spatially, according to the visual stimulus at different locations of the visual field. In previous work, we have examined these saliency operators in detail, and shown that they can significantly affect recognition performance [65]. The saliency measure of (IV.4) was shown to achieve the best performance in a number of experiments. A second possibility is to perform global feature selection. This consists of weighting each feature by how much it contributes to the recog-
nition [153, 172, 34, 118], or simply selecting the most informative features for recognition [57]. Finally, it is possible to rely on a *richer set of features*. Proposals in this area range from the use of feature extraction algorithms, such as independent component analysis (ICA) [84] or a predictive sparse decomposition [80], to the simple adoption of a large set of random features [80]. In this case, there could be a very large number of channels in each network layer. [123] has shown that a single layer network with many channels can outperform hierarchical networks with few channels per layer.

By definition, the HDSN implements the first form of (spatial) feature modulation. For the experiments of this section, the network was extended so as to implement the other two. The filter pool was first augmented with a set of 63 discrete cosine transform (DCT) filters of size $8 \times 8$ (the DCT basis set minus the average -DC- filter). This expansion was inspired by [94], who showed that a set of random projection filters can lead to better performance than a Gabor decomposition. The DCT is a proxy for such projections. Global feature modulation was then implemented with a biologically plausible feature selection mechanism. This consisted of pooling the risk of (III.13), across the visual field, for each feature $X$. The 4 channels of largest risk were then selected, and the remaining discarded. This tuned the feature set to the recognition problem of interest, while maintaining the dimensionality of the first layer identical to that of the standard HMAX model [117]. The recognition accuracy of the resulting network is presented as type V in the Table IV.5, where DS* should be read as “DS with feature selection”. Selecting simple features in Layer 1 has a gain of 2.2% in N15 and 0.7% in C101.

**Comparison to previous methods**

Table IV.6 summarizes the best results in the literature for various methods. These relate to different numbers of layer 2 units, which are also reported, and are grouped by network dimensionality. The first row of each method in the table
Table IV.6  Comparison to state-of-the-art approaches in computer vision with number of layer 2 units used.

<table>
<thead>
<tr>
<th>Method</th>
<th># Layer 2 units</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N15</td>
<td>C101</td>
</tr>
<tr>
<td>SPMK [93] (L=0)</td>
<td>400</td>
<td>200</td>
</tr>
<tr>
<td>SPMK [93] (L=2)</td>
<td>-</td>
<td>4,200</td>
</tr>
<tr>
<td></td>
<td>8,400</td>
<td>-</td>
</tr>
<tr>
<td>kNN-SVM [177]</td>
<td>-</td>
<td>3,030</td>
</tr>
<tr>
<td>V1 model [123]</td>
<td>-</td>
<td>4000</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>86K</td>
</tr>
<tr>
<td>HMAX [117]</td>
<td>-</td>
<td>4,075</td>
</tr>
<tr>
<td>NBNN [12]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>10 M</td>
</tr>
<tr>
<td>ScSPM [174]</td>
<td>450</td>
<td>5120</td>
</tr>
<tr>
<td></td>
<td>21,504</td>
<td>21,504</td>
</tr>
<tr>
<td>convNN [80]</td>
<td>-</td>
<td>4,096</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HGMM [181]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>46,080</td>
<td>310,272</td>
</tr>
<tr>
<td>HDSN</td>
<td>450</td>
<td>4,040</td>
</tr>
<tr>
<td></td>
<td>22,500</td>
<td>20,200</td>
</tr>
</tbody>
</table>
reports to “small” networks (around 400 units for N15 and 4,000 for C101), and the second row of each method to “large” (around 20,000 units for both datasets). The methods are ordered by publication year, with the earliest at the top. The table supports a number of conclusions. First, while there have been recent improvements in recognition accuracy, these are mostly due to increased network sizes, not superior recognition architectures. Note that, when the networks are calibrated for size, the recent methods do not necessarily outperform earlier ones. For example, it is not at all clear that the latest generation of the SPMK architecture (ScSPM) improves on the original (64.8% vs. 64.6% on C101, 80.3% vs. 81.4% on N15). On C101, if HDSN is ignored, the best “small” model is still the kNN-SVM. On N15, the more recent ScSPM outperforms the earlier SPMK (L=0), but the difference is not very significant. The significant leap in performance occurs from the left to the right halves of the table, i.e. from smaller to larger networks. This shows that a precise calibration of network sizes is necessary whenever different methods are compared. Second, some of the methods with best performance, e.g. NBNN or HGMM, have very large model complexity. It is not clear how well these architectures would fare under a fairer comparison that penalized network size.

Third, the HDSN achieves the clear best performance. Among small networks, it is far superior to the next best methods (82% vs. 75% on N15, 70% vs. 66% on C101). In fact, in N15, the small version of HDSN outperforms the large versions of SPMK, NBNN, and ScSPM. In C101, it is only outperformed by the large versions of HGMM, and ScSPM. The large HDSN network has the best overall performance, with the best results on N15 and results comparable to the best in C101. On N15, the only comparable network (HGMM) has ten times the number of units. Finally, among networks of equivalent size, only ScSPM is comparable to HDSN. Its large network has a recognition rate 4.2% smaller on N15, and equivalent performance on C101. For the small networks the difference is 6.5% on N15 and 5.2% on C101.
IV.E.4 Comparison to saliency measures

The results above show that the HDSN outperforms previous recognition approaches from both the biological computer vision literatures. It could be argued that this is due to the fact that these use 1) no saliency, or 2) bottom-up saliency measures, such as SIFT. The next question is how discriminant saliency fares against other top-down measures, such as the SalBayes criterion of [39]. Like discriminant saliency, this is a model of top-down saliency that combines global and localized feature modulation. Since no software is available for this criterion, we compared the two approaches on the dataset where SalBayes was originally evaluated: the Amsterdam library of object images (ALOI). This contains 108,000 images from 1,000 object categories, under 12 illumination colors, 72 illumination directions, and 72 viewpoints. The experiments are designed to test object recognition performance as a classification problem and localization performance in different setting.

Object recognition

Object recognition experiments followed the experimental setup of [39], where results are presented as a function of the number of training images. Figure IV.8 compares the recognition rates of the HDSN (1,000 units), and ScSPM (1,024 units), to those presented in [39] for SalBayes, HMAX (1,000 units), and the SIFT-based image matching method of Lowe [105]. The HDSN again outperforms all methods. With 27 training images per class, it has a recognition rate of 95.6%, while ScSPM achieves 91%, SalBayes 83.8%, HMAX 83.4%, and SIFT 72.7%.

Object localization experiments

As suggested in [39], $N \times N$ grid pattern image was created by taking random objects from the ALOI dataset with random transformations. We use
Figure IV.8 Classification performance comparison on ALOI database as a function of training size for various methods.

smaller images (92 × 76) to make the grid instead of 256 × 256. We use 25% of examples as training and use the rest as testing examples.

Each grid is used as one field of view at a time. Saliency network tuned to the target object is applied and produce $N \times N$ saliency map. In each grid the output $C_2$ vector in Figure IV.1 (a) is summed and produces one final saliency value. In this experiment, weights of each $C_2$ output element are all set to 1 and second layer patch filters which has $\alpha_0 > \alpha_1$ in equation III.2 are discarded to make each saliency output in $C_2$ indicate the presence of the given patch.\(^2\) We use 20 random patch filters per object category and the number of used patch filters is varied from 9 to 20 depending on the number of patches having passed $\alpha_1 > \alpha_0$ test. Figure IV.9 shows an example of $5 \times 5$ grid of objects and its 25 saliency maps for each target from top left to bottom right. Saliency maps show the different focus of attention according to different target. Most of the target saliency map has highest value for the matched target location. It also shows degree of confusion between similar looking targets. For example, object (first row, second column)

\(^2\)For generalized Gaussian features, feature absence produces a narrow GGD close to a delta function, while feature presence increases the variance of distribution. Therefore, when two conditional distribution share the shape parameter, $\beta$, simple variance comparison indicates the presence/absence of the feature in target class.
Figure IV.9  Examples of object categories and its saliency map. 25 saliency maps in the bottom show saliency for 25 target classes starting top left to bottom right object.
Figure IV.10  Object localization result averaged over 1000 targets. (a) precision-recall curve (b) ROC curve

Table IV.7  Percentage of scenes where the target object is located within the first saliency

<table>
<thead>
<tr>
<th></th>
<th>2 × 2</th>
<th>5 × 5</th>
<th>9 × 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>correct first fixation rate in [39]</td>
<td>60</td>
<td>38</td>
<td>–</td>
</tr>
<tr>
<td>saliency network</td>
<td>93</td>
<td>78</td>
<td>68</td>
</tr>
</tbody>
</table>

and object (second row, second column) are confused the most and shows high saliency for both of the target cases.

IV.E.4 shows the quantitative comparison results with the performance of [39]. The table shows the percentage of target which is caught with first saliency for $N \times N$, $N \in \{2, 5, 9\}$ grid search. Saliency map from proposed network shows much higher accuracy over the saliency model in [39] and it correctly localizes the 68% of 9 × 9 scenes with the highest saliency. As one more experiment, we ranked all the test images by its saliency for each target and plot precision-recall and ROC curve. The plot is averaged over 1000 targets. Note that the number of images per class is the same and the chance level is 0.001. Figure IV.10 (a) is precision-recall curve and (b) is ROC curve. The average percentage that highest saliency comes from the right target over 1000 classes is 90%. Area Under ROC Curve (AUC) is 96.7.
IV.F Conclusion

As shown in section IV.C, many state-of-the-art approaches can be mapped into the two layer network of Figure IV.1. We compare the performance of saliency network with the performance of computer vision algorithms and other saliency models under various experimental setting on multiple databases. When compared to these networks, the proposed one has the following fundamental differences:

1. a top-down definition of saliency, which equates saliency to visual stimuli discriminant for classification, vs. a bottom-up definition that equates saliency to orientation dominance.

2. discriminant selection of the layer 1 features most informative for classification

3. a second layer of discriminant saliency, which extracts saliency patterns discriminant for the target, rather than simply matching saliency patterns to stored templates.

Experimental results show that discriminant saliency network tuned to the given top-down object recognition problem in every level of network achieves better performance. More importantly, top down definition of saliency achieves robust recognition performance even the target is distinctive as absence of low level features, where most of gradient based methods are broken.
IV.G Appendix: Detailed implementation of Hierarchical Discriminant Saliency Network

IV.G.1 Image preprocessing

First, input (gray) image, $I(x, y)$, is spatially normalized. A $9 \times 9$ normalization window is used throughout all the experiments.

$$N(x, y) = \frac{I(x, y)}{\sqrt{\sum_{a=x-4}^{x+4} \sum_{b=y-4}^{y+4} I(a, b)}}$$  \hspace{1cm} (IV.27)

Normalized images are decomposed into multiple, $S$ scale image channel where scale factor is $2^{(s-1)/4}$ where $s \in 0, 1, 2, ..., S-1$. $S$ was set to 5. 'CubicFilter' option in ImageMagick is used to resize images. Scale 0 is applied to generate the next scale 1 and scale 1 is applied to generate scale 2 and so forth. Now resulting 5 scale image pyramid $N(x, y, S)$ is convolved with 4 orientation gabor filter bank with size $11 \times 11$,

$$G(x, y) = \exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda}X\right)$$  \hspace{1cm} (IV.28)

where $X = x \cos \theta - y \sin \theta$ and $Y = x \sin \theta + y \cos \theta$. $x$ and $y$ vary between $-5$ and $5$ and $\theta$ has $\{0, \pi/3, 2\pi/3, \pi\}$. The response of a patch of pixels $X$ to a particular $G$ is given by

$$R(X, G) = \left| \frac{\sum X_i G_i}{\sqrt{\sum X_i^2}} \right|$$  \hspace{1cm} (IV.29)

The multi scale response for four orientation channels is the input to simple cell in layer 1.

IV.G.2 Parameter learning in multiclass classification

We fixed shape parameter $\beta$ as .5 in General Gaussian Density (GGD)

$$P_X(x; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-\left(\frac{|x|}{\alpha}\right)^\beta}$$  \hspace{1cm} (IV.30)
and we use this value through all experiments. Only scale parameters, \( \alpha_i \), where \( i = 1 \) for target and \( i = 0 \) for non target, need to be estimated using the following estimate from training set.

\[
\hat{\alpha}_{MAP}^\beta = \frac{1}{\kappa} \left( \sum_{j=1}^{n} |x_j|^\beta + \nu \right), \quad \text{with} \quad \kappa = \frac{n + \eta}{\beta}, \quad (IV.31)
\]

where \( \nu \) and \( \eta \) are set to 1.

In multiclass object classification, the parameters are learned under binary classification setting per object class. Per each object class, the target becomes the class of interest and non target becomes all object classes excluding the class of interest. \( \alpha \) parameters for both layers assume one (target) vs the others (non target) setting. Second layer patches are also sampled in this binary classification setting which means \( N \) number of patches are randomly sampled from target and it learns \( \alpha_1 \) from target and \( \alpha_0 \) from the others. Therefore the overall sampled patches become \( M \times N \) where \( M \) indicates the number of object classes.

**IV.G.3 SVM classifier**

SVM classifier is applied to the output of the network, which has \( 1 \times P \) dimension where \( P \) is the number of patches from all targets. Input data is sphered before classification: the mean and variance of each dimension are normalized to zero and one respectively. Linear SVM is used for all the experiments with parameter \( \gamma \) is set to the median among positive values in \( N \times P \) training matrix where \( N \) indicates the number of training images. Mode is set to “1-v-all”, kernel type is set to “linear” and we use 0.0078125 as “C” parameter.

**IV.H Acknowledgments**

The text of Chapter IV, in full, is based on the material as it appears in: Sunhyoung Han and Nuno Vasconcelos, “Hierarchical Discriminant Saliency Network for object recognition”, submitted to IEEE Transactions on Pattern
Analysis and Machine Intelligence, 2011. The dissertation author was a primary researcher and an author of the cited material.
Chapter V

Application: Amorphous object detection in the wild
V.A Introduction

Many object detection approaches have been proposed over the last decade. These range from the ubiquitous bag-of-features model [178, 164], to algorithms based on representations of shape [10, 46, 49], and models of configurations of parts [45, 32, 50], among others. The development of these algorithms is, in non-trivial part, guided by the portfolio of datasets available to compare different approaches. This portfolio has been considerably enriched and diversified since the early days of the UIUC carside and Caltech4 datasets. Recent benchmarks are much more extensive, covering a much larger number of object categories, viewpoints, and intra-class variation.

Nevertheless, it could be argued that current datasets only cover the extremes of the spectrum of scenarios faced by a practical object detector. One of these extremes corresponds to the detection of broad object categories, such as “airplane”, “dog”, “cat”, etc., and datasets such as Caltech256, or PASCAL VOC. Each of the categories in these datasets is comprised of many distinguishable object sub-groups of widely different appearance, e.g. the subclasses “siberian husky”, “bulldog”, and “yorkshire terrier” of the “dog” class. This constrains the number of training examples per sub-class, which is usually small, and bias the benchmarks towards specific classifier architectures. For example, because kernelized SVMs can store large portions of the training set as support vectors, it is not surprising that the combination of these classifiers with sophisticated kernels, and the loose bag-of-words representation, achieves good performance on these datasets.

On the other end of the benchmarking spectrum are tasks such as face or pedestrian detection. These refer to narrowly defined object classes, e.g. frontal faces or upright pedestrians, and datasets with a large ratio of training images to distinguishable object sub-classes. Such datasets sample the appearance of a more or less constant object in widely varying imaging contexts. It is thus not surprising that their most successful solutions are based on templates of object appearance,
Figure V.1 Example images from the PandaCam dataset. Note the high variability of view points, illumination, and object pose.
using shapelets [133], configurations of HOG [32, 45, 107], or Haar features [167].

In this work, we investigate the detection of a large class of objects not covered by these two scenarios. These are objects that lack many of the features commonly used as cues for object detection, and which we denote *amorphous*. Strictly speaking, amorphous objects have no distinctive edge configurations, texture, or a well defined shape. They can be found in science fiction movies, in the form of jelly-like creatures that can take any desirable shape. While, in the real world, truly amorphous objects are rare, many real objects are close to amorphous (e.g. a jellyfish, a bean bag, etc.), and an even larger set *quasi-amorphous*. By this, we refer to objects that can have very characteristic appearance under some canonical poses, but appear amorphous under others. Many such examples exist in the animal world. This is illustrated by Figure V.1, which shows a Panda bear under multiple poses. While the Panda face is very iconic, faceless poses tend to be quite amorphous. We will not worry further about these fine distinctions, simply referring to such objects as *amorphous*.

This work makes two contributions to the detection of amorphous objects. The first is an object detector, based on the idea of *discriminant saliency templates*. The intuition is that (at least in the natural world) the most distinctive property of amorphous objects is their lack of low level features, when compared to the surrounding scenes. This suggests modeling these objects as *blobs of feature absence*, i.e. regions where features that are usually active for natural images have a much weaker object response. One possibility for amorphous object detection is thus to rely on *discriminant blob detection*, by identifying blob-like regions in the responses of a set of *features that are discriminant* for object detection. A potential problem is, however, that discrimination may be due to the *absence* of feature responses. We overcome this problem by formulating blob detection as a form of top-down discriminant saliency [57].

Under this approach, detection is based on a two level classification architecture which implements a combination of *feature-based* and *template-based*
discriminant saliency. The first level consists of a feature-based top-down saliency model, tuned for the detection of the target object. It is a robust classifier, that can detect the absence of a set of features, if this absence is informative of object presence. However, it is not highly selective, frequently generating false positives in background image regions. The second level learns discriminant templates of saliency response, which are then used to detect blobs of saliency compatible with the target object. This is again implemented with a top-down discriminant saliency model, tuned for object detection, which operates on saliency templates rather than image features. Altogether, this classifier is selective, yet robust enough to detect highly deformable objects of reduced visual structure. The use of saliency, rather than appearance, templates also makes it robust to pose variation.

The second contribution of this work is a dataset for the evaluation of amorphous object detectors. This dataset is assembled from video of a real animal habitat, the Panda bear exhibit of the San Diego Zoo [1]. It resembles current pedestrian datasets, in that it requires the detection of a few objects under various imaging contexts. On the other hand, it is similar to object category datasets, in the sense that Pandas have wide variability of appearance. As can be seen from Figure V.1, this is in part because they are highly deformable objects, and in part because the video feed is collected from multiple cameras, with multiple fields of view (varying backgrounds), at different distances from the Panda exhibit (varying scales), from different angles (varying poses), at different locations (indoors vs outdoors), at multiple times of the day, week, and year (different atmospheric conditions, variable shading, lighting, etc.), and with different potential occluders.

One of the attractives of the PandaCam dataset, is that it challenges currently prevalent beliefs about object recognition. For example, results on current datasets suggest that normalized representations of local image orientation are critically important for object detection. In fact, these representations are the only unifying link between the success of bag-of-features (almost invariably based on
Table V.1 Edgeness statistics for object and background in the PASCAL VOC and PandaCam datasets.

<table>
<thead>
<tr>
<th></th>
<th>plane</th>
<th>bicycle</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>.0111</td>
<td>.0176</td>
<td>.0135</td>
<td>.0153</td>
<td>.0181</td>
<td>.0164</td>
<td>.0202</td>
</tr>
<tr>
<td>back</td>
<td>.0045</td>
<td>.0107</td>
<td>.0101</td>
<td>.0104</td>
<td>.0085</td>
<td>.0103</td>
<td>.0111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>motorbike</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>.0092</td>
<td>.0112</td>
<td>.0167</td>
<td>.012</td>
<td>.0106</td>
<td>.0134</td>
<td>.0181</td>
</tr>
<tr>
<td>back</td>
<td>.0079</td>
<td>.0088</td>
<td>.0126</td>
<td>.0077</td>
<td>.0092</td>
<td>.0135</td>
<td>.0097</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>PandaCam</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>.0113</td>
<td>.018</td>
<td>.0174</td>
<td>.0076</td>
<td>.0151</td>
<td>.0113</td>
<td>.011</td>
</tr>
<tr>
<td>back</td>
<td>.0078</td>
<td>.0112</td>
<td>.0132</td>
<td>.0064</td>
<td>.0112</td>
<td>.0076</td>
<td>.0146</td>
</tr>
</tbody>
</table>

SIFT) on the PASCAL end of the spectrum, and template-based approaches (usually based on HOG) on the pedestrian end of the spectrum. On PandaCam, a comparison of the proposed detector with an equivalent approach built on templates of SIFT response shows that saliency templates achieve substantially higher localization performance. As an object detector, the proposed approach is also shown to achieve better performance than state-of-the-art methods for template-based detection, namely the discriminant parts-based model of [45], detection based on the bag-of-features model [93], the sparse coded spatial pyramid matching method of [174], and the Viola Jones detector [167].

V.B The PandaCam dataset

We start by introducing the PandaCam dataset, so as to motivate the challenges of amorphous object detection.

V.B.1 The dataset

The video feed provided by San Diego Zoo depicts the real time movement of a Panda family in a natural habitat, which includes bamboo trees, ponds, a small cave formed of rocks, and several other small structures. The dataset is divided into 5,018 positive images containing Pandas and 2,987 negative images.
without the animals. A bounding box is provided as detection ground truth for the positive images. The relative size of the objects varies from 2% to 90% of the image size. The variations in appearance are very large, due to the highly deformable shape of the Pandas, and the collection of the video from multiple cameras and multiple viewpoints. Illumination changes are also dramatic, since the dataset reports to a live cam that operates continuously, 24/7. Finally, the dataset is unique in the sense that the background clutter is much more structured than the objects to detect. Background trees, tree branches, rocks and leaves all have a rich combination of structure, shape, and texture. This is unlike the Pandas, which are mostly textureless and lack shape-defining edges.

\section*{V.B.2 Amorphous object statistics}

To demonstrate this point, statistics of the PandaCam dataset were compared to those of PASCAL VOC. In particular, we considered a measure of “edgeness”, and compared the relative amounts of this property in object and background, for the two datasets. To quantify edgeness, we filtered the image with a set of band-pass filters (Gabor functions of four orientations). It is well known that the responses $X$ of such filters to natural images follows a generalized Gaussian distribution (GGD) \cite{35}

$$ P_X(x; \alpha, \beta) = \frac{\beta}{2\alpha \Gamma(1/\beta)} e^{-\left(\frac{|x|}{\alpha}\right)^\beta}, \quad (V.1) $$

where $\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}dt$, $t > 0$ is the Gamma function, $\alpha$ a scale parameter, and $\beta$ a parameter that controls the shape of the distribution. It is also known that $\beta$ tends to be fairly stable, assuming values in the range $\beta \in [0.5, 0.8]$ \cite{145}. We have confirmed this observation, and use $\beta = 0.5$ throughout this work.

Given a training sample $D = \{x_1, \ldots, x_n\}$ of filter responses, the MAP estimate of the scale parameter $\alpha$ based on a conjugate prior is

$$ \hat{\alpha}_\text{MAP}^\beta = \frac{1}{\kappa} \left( \sum_{j=1}^n |x_j|^{\beta} + \nu \right), \quad \text{with } \kappa = \frac{n + \eta}{\beta}, \quad (V.2) $$
where $\eta$ and $\nu$ are prior hyper-parameters [56]. The scale $\alpha$ is proportional to the variance of the responses, and a good measure of their activity. For the features considered here, it measures the image edgeness along the feature orientation.

V.A presents the $\alpha$ estimates obtained for object and background using the bounding boxes provided by PASCAL VOC and PandaCam. Larger values of $\alpha$ imply more edge structure. Note that the edgeness of the Panda object is much smaller than those of most object classes in PASCAL. The Panda detection problem is unique in the sense that the background has much richer structure than the object itself.

V.C Discriminant saliency

The amorphous object detector proposed in this work is based on discriminant saliency. We next briefly review how this saliency principle can be used to implement a top-down measure of saliency, tuned for object detection. Saliency is formulated as optimal (in the minimum probability of error sense) classification of the visual stimulus into one of two hypotheses: a target ($Y = 1$) hypothesis of stimuli that are salient, and a null ($Y = 0$) hypotheses containing background stimuli [56]. Salient locations are those where target presence can be declared with largest confidence. Confidence is measured by the strength with which visual features in a region $\mathcal{A}(l)$, surrounding a location $l$, can be declared observations from the target class, by the optimal decision rule for target/background classification.

This is measured by the expected ratio of the likelihood of the observations under the target and null hypotheses, or Kullback-Leibler divergence,

$$S(l) = \int_{x \in \mathcal{A}(l)} P_{X|Y}(x|1) \log \frac{P_{X|Y}(x|1)}{P_{X|Y}(x|0)} dx.$$  

Using the standard approximation of risk by empirical risk,

$$S(l) \approx \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}(l)} \delta_i \log \frac{P_{X|Y}(x_i|1)}{P_{X|Y}(x_i|0)} \quad (V.3)$$
where $\delta_i = 1$ if $x_i$ is a sample from the target class and $\delta_i = 0$ otherwise. The class-assignment variables are inferred with recourse to the Bayes decision rule, i.e. by replacing $\delta_i$ with

$$\hat{\delta}_i = \begin{cases} 
1 & \text{if } P_{Y|X}(1|x_i) \geq 0.5 \\
0 & \text{otherwise.}
\end{cases}$$

With these estimates, (V.3) can be written as [65]

$$S(l) = \frac{1}{|A|} \sum_{l \in A(l)} \xi(p(l)) \quad (V.4)$$

$$\xi(x) = \begin{cases} 
\frac{1}{2} \log \frac{x}{1-x}, & x \geq 0.5 \\
0, & x < 0.5,
\end{cases}$$

where

$$p(l) = P_{Y|X}(1|x(l))$$

is the posterior probability of the target class given the feature responses observed at location $l$. This posterior probability is itself a function of the log-likelihood ratio, since

$$p(l) = \sigma[g(x(l))] \quad (V.5)$$

with

$$g(x(l)) = \log \frac{P_{X|Y}(x(l)|1)}{P_{X|Y}(x(l)|0)} \quad (V.6)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (V.7)$$

When feature responses have GGD distribution of scales $\alpha_1$ and $\alpha_0$ under the target and null hypotheses, respectively, the log-likelihood ratio simplifies to

$$g(x(l)) = \frac{|x(l)|^\beta}{\alpha_0^\beta} - \frac{|x(l)|^\beta}{\alpha_1^\beta} + T, \quad (V.8)$$

where $T = \log \left( \frac{\alpha_0}{\alpha_1} \right)$. The parameters $\alpha_i^\beta$ are learned from training samples $D_i$ of feature responses from the target and background classes, using (V.2).
V.D The importance of feature absence

Feature absence has an important role in the detection of amorphous objects. We next discuss how discriminant saliency naturally accounts for this.

V.D.1 Amorphous objects and feature absence

Amorphous objects lack many of the features that are commonly used as cues for object detection. They do not have many distinctive edges, may not have a very distinguishable texture, and are characterized by a large shape variability. In fact, as shown in Figure V.1, they can be thought of as blobs of low image complexity. However simple blob detection [23, 173] is unlikely to successfully find Pandas, as there are many blob-like regions in the backgrounds of Figure V.1: smooth rocks, tree trunks, light reflections on interior walls, areas of the exhibit floor, etc. One possibility is to rely on discriminant blob detection, by identifying blob-like regions in the responses of features that are discriminant for Panda detection. These are regions of feature absence, i.e. where features that are usually active for natural images, have a much weaker response to the Panda stimulus. Feature absence is naturally detected by discriminant saliency.

V.D.2 Saliency and feature absence

We start by noting that this is not true for all formulations of saliency. While many saliency detectors have been proposed in the literature, most emphasize the detection of the presence of certain features in the visual field. This is, for example, the case of interest point detection, which is explicitly formulated as the detection of corners [66], points of image curvature [112], activity [83], or texture complexity [35]. A popular generalization of this idea [130, 180, 18], is to detect features of low probability within the visual field. Its decision theoretic implementation, for GGD feature statistics, is to find the locations $l$ where feature
responses have maximum entropy [83]

\[ H(l) = -\int_{A(l)} P_X(x) \log P_X(x) dx \quad (V.9) \]

\[ = K + \frac{1}{2|A(l)|} \sum_{i \in A(l)} \left( \frac{|x_i|}{\alpha} \right)^\beta \]

where \( K = \log 2 \alpha \Gamma(1/\beta)/\beta \), and \( \alpha \) is the GGD scale for the feature responses across the visual field. Note that this is, in many aspects, similar to the measure of (V.3). It is, in fact, equivalent in the limit of \( \alpha_1 \to \infty \), if \( \alpha = \alpha_0 \) and the non-linearities \( \sigma(.) \) and \( \xi(.) \) are replaced by the identity map. While this may appear a small difference, it has a major impact on the ability of the saliency detector to switch between the detection of feature presence and absence.

Discriminant saliency can switch between these two detection strategies because it has access to two scale parameters, \( \alpha_1 \) and \( \alpha_0 \). When \( \alpha_1 = \alpha_0 \), the target and null distributions are identical, i.e. there is nothing to discriminate, and saliency is null for all \( x \). When \( \alpha_1 > \alpha_0 \), the target distribution has a heavier tail than that of the null hypothesis, and saliency is high for large feature responses. Conversely, the null hypothesis has heavier tail when \( \alpha_1 < \alpha_0 \), and only small feature responses are salient in this case.

The two behaviors are illustrated in Figure V.2 a), which presents the characteristic function of the discriminant saliency detector - curve of \( \xi(P(l)) \) as a function of \( x(l) \) - for different values of \( \alpha_1 \), when \( \alpha_0 = 1 \). Note how 1) the presence of the feature \( X \) (large feature response \( |x| \)) elicits a strong saliency response when \( \alpha_1 > 1 \), but 2) strong saliency responses are reserved for feature absence (small \( |x| \)) when \( \alpha_1 < 1 \). For comparison, the characteristic curve of the entropy-based detector is shown Figure V.2 b), for \( \alpha = 1 \). In this case, all degrees of freedom are exhausted once the background distribution is fixed, and the saliency detector is always a detector of feature presence.
V.E Amorphous object detection

We next consider the design of an amorphous object detector based on saliency templates, derived from the discriminant saliency principle.

V.E.1 Discriminant saliency networks

The saliency computations of Section V.C can be implemented with a network that mimics the standard neurophysiological model of the visual cortex [65]. The network has two layers, one of simple and another of complex cells. The simple cell layer computes the target posterior map $\mathcal{P}(l)$, which is then transformed into the saliency map $\mathcal{S}(l)$ by the complex cell layer. A simple cell is associated with each location $l$ of the visual field, and implements the computations of (V.5)-(V.8) and (V.2). This is the combination of filtering, divisive normalization, and a saturating non-linearity with which simple cells are modeled under the standard neurophysiological model. A complex cell then pools the simple cell responses within the region $\mathcal{A}(l)$, after application of the non-linearity $\xi(x)$, to implement (V.3). These are the operations of complex cells under the standard neurophysiological model. The resulting saliency value, $\mathcal{S}(l)$, is a decision-theoretic measure of the confidence with which the feature responses at $l$ can be assigned to the target class.
The two layer network of (simple and complex) units is denoted a discriminant saliency network [65].

V.E.2 Discriminant saliency templates

The proposed amorphous object detector is based on discriminant saliency templates. These are discriminant templates of saliency response, derived from features that are themselves discriminant for the detection of the target object. As illustrated in Figure V.3, the detector is implemented as a two-stage discriminant saliency network. The two stages are identical up to the features used to evaluate saliency, i.e. the linear filtering implemented by their simple cells. The first layer relies on low level features, such as measures of image orientation [138, 117], color opponency, image intensity [39], projections into standard signal basis (wavelets or Gabor expansions), or even random projections [94]. These features can be deemed discriminant due to either their presence or absence in the target object. In this work, we use the first four filters in a discrete cosine transform (DCT) basis of size $8 \times 8$, other than the average (DC) filter. These filters are illustrated in Figure V.3. For amorphous objects, saliency responses usually reflect a mix of feature presence and absence, allowing the saliency map to be active in blob-like regions of low image complexity.

The second stage aims to detect configurations of saliency, produced by the first, which are distinctive for the target object. These salient configurations of salient feature responses capture information about object shape. The implementation of the second network stage has some resemblance to the second layer of the HMAX network of [138, 117]. A number of saliency templates are first randomly sampled from the outputs of the first network stage, during training. This is done by extracting patches, centered at random locations and scales, of response to random target images. Each patch has dimension $n \times n \times 4$, for $n \in \{4, 8, 12, 16\}$, and is normalized to zero mean and unit norm (over the 4 channels). During detection, the normalized patches (denoted patch filters) are correlated with first stage
Figure V.3  Architecture of the proposed template-based saliency detector. DCT features are used in the first network layer, and saliency templates in the second. The saliency computation is repeated at multiple scales.
responses, to extract a second level of feature responses. These are then processed by the second stage of the network, to determine the saliency of these responses. Note that, unlike the HMAX network, which measures distances between patch filter and first stage response, this operation measures the discriminant power of the patch filter (saliency template) to classify first stage responses into object and background. This reinforces the responses of the templates that are discriminant for object detection and suppresses those of templates that are not.

V.F Experiments

In this section, we describe a number of experiments designed to evaluate the performance of the proposed amorphous object detector.

V.F.1 Experimental Setup

In all experiments, 2,518 positive images were used for training and the remainder 2,500 as a test set. All positive training examples were cropped and normalized to a height of 80 pixels, while maintaining the original aspect ratio. The assembly of negative examples followed the iterative procedure of [32, 45]. Object detection was based on the saliency maps produced by the network of Figure V.3. 5,000 templates were randomly selected from first layer responses to positive training examples. The KL divergence between the GGD responses to object and background was then computed per template. The 500 templates of largest discriminant power were finally selected. Saliency maps produced by these templates were added into an overall saliency map, which was used for object detection.

Object detection was performed at 7 scales of a pyramid decomposition of each test image. More precisely, an image of size $H \times W$ was expanded into 7 pyramid layers of size $2^{0.5i}H \times W$, for $i \in \{-1, 0, 1, \ldots, 5\}$. This produced 7 saliency maps per image. The location of largest saliency was then found, at each scale,
with a combination of box filtering and non-maximum suppression. A box filter of size $N \times N$ and amplitude $1/(N \times N)^\gamma$, was first convolved with all scale saliency maps, using $N = 80 \times 2^{0.5i}$ for scale $i$. The parameter $\gamma$ was determined by cross validation. Non maximum suppression was then applied to the filtered saliency maps, to detect the location of largest saliency. The scale of largest saliency was finally selected.

V.F.2 Saliency as focus of attention

We start by analyzing the performance of template-based saliency as a focus of attention mechanism. Its localization performance is compared to those of a SIFT-based saliency method, and a localization method based on discriminant visual words [36]. The SIFT-based saliency method is identical to that now pro-
posed, but uses templates of SIFT response instead of saliency templates. Each image is represented by a collection of SIFT descriptors, extracted on a dense sampling grid of $16 \times 16$ patches, with 6 pixels of grid spacing. As for template saliency, 5,000 templates of SIFT response were randomly chosen from the set of responses to positive examples, and normalized to zero mean and unit variance. Theses SIFT templates were then correlated with the SIFT responses to each training example, and the KL divergence between responses to object and background computed per template. The most discriminant 500 templates were finally used to produce SIFT-based saliency maps.

To compute saliency from visual words, images are represented as bags of SIFT descriptors, and quantized with a codebook of 1,000 words, learned with k-means. The discriminant power of visual word, $w$, is then measured by the discriminability function proposed in [36],

$$D(w) = \frac{\# \text{ target images containing } w}{\# \text{ images containing } w}.$$  \hfill (V.10)

The SIFT descriptor extracted from each image location $l$ is then quantized into the closest visual word $w^*$. The saliency at $l$ is the discriminability $D(w^*)$.

Figure V.4 shows examples of saliency maps produced by the three methods. Test images are shown on the top row, template-based saliency maps on the second, SIFT-based saliency maps on the third, and saliency maps based on visual words on the fourth. Note that the latter are very noisy, with many false positives on the background, and few strong responses at target locations. SIFT-based saliency maps have much better localization, suppressing most responses from the background. However, while capturing the edge or contour structure of the target, they fail to respond to the object interior. This is not surprising, since SIFT is based on image gradients and the object interior is mostly smooth. Nevertheless, it is quite difficult to locate the object from these saliency maps. This task is much easier from the saliency maps produced by saliency templates, which 1) are active in the object interior, and 2) have even less false positives on the background. An objective comparison of localization performance in given in
Figure V.5 Precision recall curves for object localization. We compare localization performance of template saliency, SIFT saliency, and discriminant visual word.

Figure V.5, in the form of a precision recall curve. This curve is produced by thresholding the saliency map at various amplitude levels, measuring the overlap between the above-threshold region and the bounding box ground truth, and averaging over test images. The average precision is 0.31 for template saliency, 0.23 for SIFT saliency, and 0.16 for discriminant visual words.

V.F.3 Detection performance

The detection performance of template-based saliency was compared to those of the discriminatively trained part based model (partModel) of [45], the sparse coded spatial pyramid matching (ScSPM) method of [174], the bag-of-features (BoF) method of [93], and the Viola-Jones (VJ) detector, which combines boosting and Haar features [167]. The partModel was learned with 6 components, and the results reported were obtained with the non maximum suppression method of [45]. Detection with ScSPM, BoF, and VJ was based on a sliding window,
Figure V.6 Examples of detections by the template-based saliency detector. White windows indicate ground truth, blue detections and red false-positives.
using windows of seven scales, and a step size of 10 pixels. The non-maximum
suppression scheme used for template saliency was also applied to these methods.
For BoF and ScSPM, we used a spatial pyramid of 2 levels and a codebook of
1,000 visual words.

Object detection performance was evaluated with the PASCAL measure,
which requires an overlap greater than 50% between the bounding boxes of the
detection area and ground truth. Figure V.7 shows the curve of detection rate
vs false positives per image (fppi) for all methods. The partModel was unable to
model the Panda with the finite set of poses those were available, and achieved the
worst performance of all methods. Both ScSPM and BoF produced a significant
improvement, with ScSPM achieving slightly better performance. Another per-
formance boost was achieved with the VJ detector. Finally, the template-based
saliency detector produced the overall best performance. The detection rate at
0.3 fppi was 71.5% for template saliency, 66% for VJ, 58.6% for ScSPM, 56.8%
for BoF and 43.8% for the partModel. Figure V.6 shows detection examples by
the template-based saliency detector. White windows indicate the ground truth
and blue and red windows indicate detections. Blue is used for correct detections
under the PASCAL measure, and red for false positives.

Besides the superiority of template-based saliency, an interesting conclu-
sion from these experiments is the good performance of VJ. This is due to the
fact that, unlike the partModel, BoF, and ScSPM, this method does not depend
on image gradients. Instead, it relies of Haar features that can capture edgeless
blobs. These results contradict the common belief on the importance of gradients
for object recognition, clearly showing that gradient based detectors are not the
answer for all detection problems.
Figure V.7 Curves of detection rate vs false positives per image. Performance is compared with state-of-the-art computer vision algorithms.

V.G Acknowledgments

The text of Chapter V, in full, is based on the material as it appears in: Sunhyoung Han and Nuno Vasconcelos, “Amorphous object detection in the wild” , in , 2011. The dissertation author was a primary researcher and an author of the cited material.
Chapter VI

Application: Automatic detection of object-based ROI (Regions of Interest) for image compression using discriminant saliency network
VI.A Introduction

With the advent of the “second-generation” image coding [91] which puts high emphasis on the selection of the messages to be coded have inspired image/video coding techniques. JPEG2000 incorporated region-of-interest (ROI) image coding [76] and MPEG4 is object-based video coding standard [2] where scene is decomposed of several video objects (VOs) that are separately encoded and decoded. Coding algorithms utilizing predefined spatial priority information have been proposed to dynamically assign bits [134, 103, 25], use scalable coding for variable bandwidth [139, 169], use different level of error protection for important region [135].

The automatic identification of these regions of interest (ROIs) is, however, a non-trivial problem, since they are determined by the perceptual mechanisms of visual attention. While these mechanisms are still poorly understood, it is known that attention is driven by two complementary mechanisms. The first, usually referred as bottom-up, is very fast, completely stimulus driven, i.e. active even when a subject is not actively pursuing a task, and does not need training. In bottom-up visual attention theory, it is assumed that there is a unique eye fixation path depending on image context which is able to have consensus from multiple subjects. Visual attention algorithms are often compared to experimental eye movement data from human subjects for validation. Since human only can attend small portion of an image knowing the path of eye fixation to a given image leads to high yield compression which only compress attended image regions. Itti [77] showed biologically plausible attention model which can predict general human eye movement and its applicability to high yield video compression in the expense of visual degradation of the region of having less attention. Algorithmic approaches which rely on known property of human visual system have been proposed based on object size, contrast, shape, color, motion, or novelty [120, 175, 146].

Bottom-up methods produce single ROI map for compression to the given
Figure VI.1 Example of image which may contain different region of interest. This image contains multiple objects such as big ben, bus, people, car, etc.

image which should explain general visual attention of different subjects for different time period. General visual attention from an image is a tricky concept and human eye tracking experiment with well instructed subject (e.g. instructions require focusing on general understanding rather than free viewing to get task independant eye movement [77]) cannot show a perfect agreement between subjects and idiosyncratic eye movements are expected for the free viewing where individuals have different task in mind. ROI definitions that cannot be made task specific, i.e. tuned to the specific application or the specific subject at the end of the image processing chain. Consider, for example, the image of Figure VI.1. While one subject may be interested in closely inspecting the architecture of the monument, another may be more interested in the trademark London taxicabs, and a third may want to identify the people present in the scene. In the absence of top-down processing it is impossible to adjust the ROI according to these goals.

On the other hand, the second, usually referred to as top-down, is slower and task dependent and lead to useful visual attention when specific task goal is involved. Top-down definition of visual attention is particular useful for image searches where the large web collections of data need to be shown in small hand-
held devices with limited display. Query based visual attention becomes new issue with side spread of smart phones and the usages for image searching on both navigational and informational purpose [17]. Landmark recognition for mobile robot navigation lines up for potential application of top-down attention [149, 136, 27].

Visual attention is defined per image basis. Algorithmic methods computationally define important image regions based on object size, contrast, shape, color, or novelty [120, 175, 146].

Designing ROI detectors which are tunnable according to either application or subject is, however, not simple. Real time interactive gaze-contingent foveation where eye-tracking device records eye position from a human observer [37, 95, 127, 128] showing promising compression rate with user specified ROI maps. However, these type of approaches have severe limitations for extension to general applications. Object based or search query based ROI problem was investigated in [102, 99, 100]. These are limited for the extension to virtually infinite object classes. Since it is a-priori unknown what subjects may be interested in, the detector should be generic enough to handle large numbers of object categories. While fast and highly accurate object detectors are currently available for some categories, such as faces and cars [166], their application to other classes is problematic. The main bottleneck is the complexity of training, which requires careful manual assembly of very large training sets, composed of precisely cropped and aligned example images. The resulting cost, in terms of both time and manual labour, makes the user-guided training of these detectors infeasible. Web image sorting algorithms have been proposed but these algorithms do not investigate object localization issues.

In this work, we pursue an alternative strategy for the design of ROI detectors. This strategy is of a top-down nature, but trades the emphasis on highly accurate classification (characteristic of classical detector design) for an emphasis on 1) weak supervision and 2) learning efficiency. It consists of two stages: saliency detection and saliency validation. In the first, given an object
class of interest, we search discriminant complex features (or filters) for the subset that best discriminates between 1) the examples from that class and 2) a set of generic images, representative of the distribution of all natural imagery. This can be done very efficiently, with the discriminant saliency network which has biological plausibility of human visual cortex and exploits the intermediate complex features coming from hierarchical architecture. Intermediate complex feature is shown to have superior performance compared to simple or more complex features [155]. Filtering an image with this discriminant complex features produce a discriminant saliency map, which has high magnitude in the locations of the class of interest and low magnitude elsewhere.

Of course, the cost of computational efficiency is some loss of classification accuracy. When compared to the output of a classical (strongly supervised) detector, discriminant saliency maps are less accurate in two ways. The first is smaller localization accuracy, i.e. ROI boundaries that are not as crisp as those obtained with strong supervision. This is not a significant problem for the applications considered here, since the relative increase in ROI area tends to be small. The second is a higher false-positive rate, i.e. the classification as salient of regions that may not contain objects of interest. This is a more significant problem, and is addressed by the saliency validation stage. This stage is inspired by recent developments in computer vision, which have shown the benefits of representing objects as constellations of “parts” [48, 4, 45]. The basic idea is to equate locations of maximal saliency with object “parts”, and build a model for the spatial configurations of these parts in the class of interest. The salient configurations of the image under analysis are then rated according to how well they are explained by this model. This imposes a constraint of geometric consistency between ROIs and the training examples, and is a powerful filter for the rejection of false-positives. We develop a computational efficient validation step by approximating all saliency maps by Gaussian mixtures, and using fast hierarchical inference procedures for model-building. While our geometric constraints are quite simple (much simpler
than those commonly used in vision [4, 48, 45]), their combination with discriminant saliency is rather effective.

Overall, because we search for both the features and configurations that are more common in the object class of interest than in the generic class of natural images, the proposed strategy is fairly robust to the presence of clutter in the training images. This makes it possible to learn without the requirement for manual segmentation or alignment of examples during the assembly of the training set. The cost of tailoring the architecture for a new class is therefore quite low, making it possible for users to define new ROI detectors. The training sets used in all of our experiments were automatically downloaded from the web, without any manual processing other than rejecting search engine errors, i.e. images that did not match the specified query. Furthermore, the architecture is completely generic, and applicable to any object category. We illustrate these properties by designing ROI detectors for image compression, in the context of various application scenarios that may be of real interest.

This is the extension of the work [64] and shows real world scenarios for ROI image compression with hierarchical saliency model. This work also provides detailed derivation for the hierarchical EM algorithm briefly shown in [64].

**VI.B Salient point detection**

Given a class of interest this is implemented with the minimum probability of error rule for the classification problem that opposes that class to the null hypothesis. The minimum probability of error rule consists of a likelihood ratio test between the two hypothesis, where the different features are weighted according to their saliency, and produces a measure of the discriminant saliency of
each image location

\[ S(l) = \sum_k I(X_k; Z) R_k(l) \]  

(VI.1)

\[ R_k(l) = \{\max[-Im(l) * F_k, 0]\}^2 + \{\max[Im(l) * F_k, 0]\}^2 \]

where \( R_k(l) \) is the channels resulting from half-wave rectification of the outputs of saliency filter \( F_k \) from feature \( X_k \). [59] We refer to \( S(l) \) as the saliency map with respect to the class of interest.

Salient points are defined as the local maxima of the saliency map. They are identified by feeding the latter to a peak detection module, based on non-maximum suppression, implemented with a winner-take-all (WTA) network [89]: the location of largest saliency is identified, the saliency map suppressed (set to zero) in a neighborhood of diameter equal to the saliency scale at that location, the next most salient location is found, its neighborhood suppressed, and so on. The saliency scale of a given location is the scale (radius of the region of support) of the most salient luminance feature at that location. The process is iterated until the entire saliency map is suppressed. The result is a table of salient points characterized by location \( l_k \), scale \( s_k \), and amplitude \( S(l_k) \). Note that salient locations are ordered by decreasing saliency. Finally, the amplitude of the salient points is normalized

\[ \theta_k = \frac{S(l_k)}{\sum_{j=1}^N S(l_j)} \]  

(VI.2)

and the detector outputs the list of salient point parameters

\[ z_k = (\theta_k, l_k, s_k)^T, k = 1, \ldots, N. \]

The number of point \( N \) is chosen so that \( S(l_k) < \max_i S(l_i) \times 0.1, \forall k = 1, \ldots, N \), therefore eliminating locations of trivially small response. The various modules are summarized in Figure VI.2.
Figure VI.2  Salient point detection. Image is decomposed into multiple scale layer followed by convolution with feature and finally generates saliency table by applying winner take all network.

VI.C  Salient point validation

The saliency detection procedure described above favors computational efficiency over detection accuracy. When applied to complex scenes, where the target visual concept is presented against a background containing substantial amounts of clutter, it can have a relatively large false positive rate. This problem is addressed with an, equally efficient, validation procedure that rejects configurations of salient points which are geometrically inconsistent with the training examples of the target class.

VI.C.1  Representation

Saliency validation is based on a probabilistic representation of the saliency map. To account for false-positives (salient locations which do not depict the target visual concept) we divide the set of salient point parameters $\mathcal{Z} = \{z_1, \ldots, z_N\}$ into two mutually exclusive subsets $\mathcal{Z} = \mathcal{Z}_+ \cup \mathcal{Z}_-$, where $\mathcal{Z}_+$ contains the salient points whose region of support overlaps with image region covered by the target visual concept and $\mathcal{Z}_-$ the remaining. We next introduce a binary variable $Y$ which denotes whether the image location under consideration depicts the target or not, i.e. $Y = 1$ if the location belongs to the region of support of the target and $Y = 0$ otherwise.

The generative model for the saliency map is as follows. First, a label is drawn from $Y$, determining whether the salient point is a true or a false positive.
If $Y = 1$ (true positive) the $i^{th}$ salient parameter vector in $Z_+$ is selected with probability $\pi_1^i$. The salient location is finally sampled from a Gaussian distribution whose mean and variance are given by the salient parameters $l_1^i$ and $(s_1^2)^1I$. If $Y = 0$ (false positive) the salient parameter vector is selected from $Z_-$, and the Gaussian distribution has parameters $l_0^i$ and $(s_0^2)^0I$. The overall model is a Gaussian mixture,

$$P_X(x) = P_{X|Y}(x|1)P_Y(1) + P_{X|Y}(x|0)P_Y(0)$$

$$= \sum_{i=1}^{N^1} \pi_1^i G(x, l_1^i, (s_1^2)^1I) + \sum_{i=1}^{N^0} \pi_0^i G(x, l_0^i, (s_0^2)^0I)$$

where

$$\pi_1^i = \frac{\frac{\theta_1^i(s_1^2)^1}{\sum_{i=1}^{N^1} \theta_1^i(s_1^2)^1 + \sum_{i=1}^{N^0} \theta_0^i(s_0^2)^0}}.$$

and $\pi_0^i$ is defined similarly. Note that the weight of each Gaussian is a function of both the saliency amplitude and scale.

VI.C.2 Generative model for saliency configuration

The elimination of false-positive salient points assumes that the visual concepts of interest have a consistent geometric configuration. A generative model is assumed for this configuration, and its parameters are learned from training data. Given the results of discriminant saliency on an unseen test image, the likelihood of the configuration of the detected points under the learned model is measured, enabling the rejection of geometrically inconsistent configurations.

The choice of generative model for the saliency configuration was influenced by three main goals: flexibility, efficiency, and discrimination. Flexibility was seen as the most important requirement. In principle, better false-positive rejection should be possible with models that more accurately account for the geometry of the target. On the other hand, very detailed models are hard to learn with accuracy, can be brittle (generalize poorly), and tend to require detailed specification of the probabilistic relationships between the different variables. Since this type of specification is beyond the reach of most naive users, the adoption of very
detailed models would imply their pre-specification, dimishing the ability to tune
the system to the diversity of visual classes that may be of interest in applications
such as image compression.

Computational efficiency was seen as a mild, but necessary, requirement. While various graphical models have been proposed in the vision literature for modeling the relationship between object parts [48, 143, 47], their learning complexity (hours if not days for relatively small training sets) makes them unrealistic for user-driven training. While the goal is not necessarily to be optimal in terms of computational efficiency, the model should be amenable to training with relatively large amounts of data in a user-tolerable amount of time (typically less than a minute). This reinforces the flexibility of the whole compression system.

In summary, both flexibility and efficiency advise the use of simple configuration models. We have, for these reasons, adopted a simple “blob-based” model, which reduces the overall saliency distribution to a Gaussian. To maximize discrimination we do, however, account for both the true and false-positive saliency classes. This leads to a configuration model of the form

\[ P_X(x) = \alpha_1 \mathcal{G}(x, \mu_1, \Sigma_1) + \alpha_0 \mathcal{G}(x, \mu_0, \Sigma_0), \]  

where \( \alpha_1 + \alpha_0 = 1 \), \( \mu_1 \) and \( \mu_0 \) are the centers of mass the true and false-positive saliency maps, and \( \Sigma_1 \) and \( \Sigma_0 \) their covariances.

VI.C.3 Parameter learning

For the configuration model, learning consists of deriving the parameters \( \alpha_j, \mu_j, \Sigma_j, j \in \{0, 1\} \) from the parameters of the Gaussian mixture that models the saliency distribution in (VI.4). We consider two possibilities. The first is that in which the salient points are labeled, i.e. it is known if each point is a true or false positive. This is an unrealistic assumption, but has a straightforward solution which we derive first. The second assumes that no labeling is available. We show that this can be solved with resort to an efficient hierarchical expectation-maximization algorithm, that generalizes the solution obtained with labeling.
Supervised learning

When the assignment of the salient point to the true-positive $Z^+$ and false-positive $Z^-$ sets is known the derivation of the configuration parameters is straightforward. Using the chain rule of expectation, from which expectation of a function of two random variables $X$ and $S \in \{1, \ldots, N\}$ is

$$E_X[f(x)] = \sum_i P_S(i) E_{X|S}[f(x)|i], \quad (VI.5)$$

it follows that, for $j \in \{0, 1\}$,

$$E_{X|Y}[x|j] = \sum_{k=1}^{Nj} p^j_k l^j_k \quad (VI.6)$$

$$E_{X|Y}[xx^T|j] = \sum_{k=1}^{Nj} p^j_k [(s^2)^j_k I + l^j_k (l^j_k)^T]$$

with $p^j_k = \frac{\pi^j_i}{\sum_{i=1}^{Nj} \pi^j_i}$. The parameters of the configuration model are, therefore, given by

$$\mu_j = \sum_{k=1}^{Nj} p^j_k l^j_k \quad (VI.7)$$

$$\Sigma_j = \sum_{k=1}^{Nj} p^j_k (s^2)^j_k I + \sum_{k=1}^{Nj} p^j_k l^j_k (l^j_k)^T - \mu_j (\mu_j)^T$$

$$= \sum_{k=1}^{Nj} p^j_k (s^2)^j_k I + \sum_{k=1}^{Nj} p^j_k (l^j_k - \mu_j) (l^j_k - \mu_j)^T, \quad (VI.8)$$

i.e. the mean of a class is the sample mean of the salient locations in that class, and the covariance is a sum of 1) the average scale of the salient points in the class with 2) the scatter of the salient locations. Finally, the class probabilities are given by

$$\alpha_j = \sum_{k=1}^{Nj} \pi^j_i. \quad (VI.9)$$

Unsupervised learning

In general, the classification of the salient points into true or false-positives is not known. In this case, the variable $Y$ is hidden and, instead of (VI.4), all that
is known is that the overall saliency distribution is a Gaussian mixture

\[ P_X(x) = \sum_{i=1}^{N} \pi_i \mathcal{G}(x, l_i, s_i^2 I). \]  

(VI.10)

The estimation of the configuration parameters of (VI.4) requires the assignment of the salient point parameters \( \theta_i = (\pi_i, l_i, s_i^2 I), i = 1, \ldots, N \) to the true and false-positive classes. This is a problem of clustering Gaussians distributions into a small number of mixture components and can be solved with the hierarchical EM algorithm presented in [162] for leaning mixture hierarchies. In the saliency context, the hierarchy has two levels, the bottom one consisting of the image saliency mixture of (VI.10) and the top one of the configuration model of (VI.4).

With respect to the algorithm presented in [162], the saliency problem introduces a slightly more complicated problem. This is due to the fact that the training images are not necessarily aligned, i.e. the salient locations \( l_i \) are not represented in the same coordinate frame. While, in the supervised case, it is acceptable to assume that the labeling process produces the alignment information (e.g. by manually indicating a reference point of the target in all images), this is undesirable when the goal is to support unsupervised model building. The various images in the training set therefore have to be aligned as part of the EM process.

**Hierarchical EM algorithm**

This is accomplished by considering the salient points extracted from each image separately. If there are \( K \) images, the training set as a collection of \( K \) salient point sets, where the \( k^{th} \) set contains \( N_k \) salient points \( \theta_{ik} = (\pi_{ik}, l_{ik}, s_{ik} I), i = 1, \ldots, N_k \) derived from the \( k^{th} \) image. It is assumed that that these points are subject to a translation of \( \mu_k \) with respect to the origin of a canonical coordinate frame and the image level saliency mixture becomes

\[ P_X(x) = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \pi_{ik} \mathcal{G}(x, l_{ik} - \mu_k, s_{ik}^2 I). \]  

(VI.11)
The configuration model is assumed to be centered at the origin of the canonical coordinates, i.e.

\[ P_X(x) = \alpha_1 G(x, 0, \Sigma_1) + \alpha_0 G(x, 0, \Sigma_0), \tag{VI.12} \]

with \( \alpha_1 + \alpha_0 = 1 \).

Given a set of image saliency parameters \( \theta_{ik}, k = 1, \ldots K, i = 1, \ldots, N_k \), the parameters of the configuration model \( \{\alpha_j, \Sigma_j\}, j \in \{0, 1\} \), and the displacements \( \mu_k, k = 1, \ldots, K \) are learned from with resort to a hierarchical EM algorithm, that iterates between the following steps (see Appendix VI.H.1 for a detailed derivation).

**E-step:** for \( k = 1, \ldots K, i = 1, \ldots, N_k \) and \( j \in \{0, 1\} \), compute

\[
h_{ik}^j = \frac{G(l_{ik} - \mu_k, 0, \Sigma_j)e^{-\frac{1}{2} \text{trace}((\Sigma_j)^{-1}(s_{ik}^2 I))}M_{ik}\alpha_j}{\sum_{l \in \{0,1\}}[G(l_{ik} - \mu_k, 0, \Sigma_l)e^{-\frac{1}{2} \text{trace}((\Sigma_l)^{-1}(s_{ik}^2 I))}M_{ik}\alpha_l}}. \tag{VI.13}
\]

**M-step:** \( k = 1, \ldots K, i = 1, \ldots, N_k \) and \( j \in \{0, 1\} \), compute set

\[
(\alpha_j)^{new} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \frac{h_{ik}^j}{N_k} \tag{VI.14}
\]

\[
(\Sigma_j)^{new} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \frac{1}{h_{ik}^j \pi_{ik}}
\]

\[
\times \left[ \sum_{i=1}^{N_k} h_{ik}^j \pi_{ik} (l_{ik} - \mu_k)(l_{ik} - \mu_k)^T + s_{ik}^2 I \right]
\]

\[
\mu_k^{new} = \left( \sum_{j=0}^{1} \sum_{i=1}^{N_k} \Sigma_j^{-1} h_{ik}^j \pi_{ik} \right)^{-1} \sum_{j=0}^{1} \sum_{i=1}^{N_k} \Sigma_j^{-1} h_{ik}^j l_{ik} \pi_{ik} \tag{VI.15}
\]

As usual for EM, the E-step computes the posterior probability of the salient points under the two classes, and the M-step computes sample statistics weighted by these posterior assignments. Note the similarity between the M-step equations and (VI.8)-(VI.9). The hierarchical EM equations are also quite similar to those of the standard EM algorithm for learning mixture parameters, but account for
the additional information contained in the image level covariances and weights. If the covariances are equal to the identity, and the weights $\pi_{ik}$ follow a uniform distribution, the algorithm basically reduces to standard EM, applied to the salient locations $l_i$.

The final difference is that, in the parameter update step, salient points from the different images are aligned by application of the displacements $\mu_k$. The new parameters are then computed for each image and averaged over the set of training images. The displacement vectors $\mu_k$ can also be seen as the centers of mass of the saliency map extracted from each image, before the image alignment. [This is similar to Josic and Frey CHECK THIS]. Ideally, they should be computed only from true-positive salient points, since these by definition cover the target. False-positives are due to background objects and have much greater variability. As illustrated by examples (b) and (d) of Figure VI.4, false-positives (black circles) can be outliers that appear substantially far from the object.

**Robust estimation**

As is usual in statistics, the presence of outliers advises the adoption of a robust estimator. In the context of EM, robustness can usually be achieved by exploiting the fact that the posterior probabilities of each class are available for each point. In the specific case of the saliency problem, we limit the set of points that contribute to the estimation of the displacements to those that can be classified as true-positives with a minimum probability of error rule, i.e. those which have greater than 0.5 probability of belonging to the true-positive class. This is implemented by modifying the M-step according to

$$\mu_k^{new} = \frac{\sum_{i=1}^{N_k} \delta[h_{ik}^1](l_{ik} - \mu_k)\pi_{ik}}{\sum_{i=1}^{N_k} \delta[h_{ik}^1]\pi_{ik}}$$  \hspace{1cm} (VI.17)

where

$$\delta[h_{ik}^j] = \begin{cases} h_{ik}^j & h_{ik}^j \geq 0.5 \\ 0 & \text{otherwise.} \end{cases}$$
Initialization

The EM algorithm is initialized with the following parameter values

\[
\Sigma_1 = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \pi_{ik} \left\{ (l_{ik} - \mu_k)(l_{ik} - \mu_k)^T + s_{ik}I \right\} / N_k
\]

\[
\Sigma_0 = I * \max(\Sigma_1)
\]

\[
\mu_k = \frac{\sum_{i=1}^{N_k} l_{ik} \pi_{ik}}{N_k}
\]

(VI.18)

Examples

Figure VI.3 illustrates the results of the learning process using an example of image of the "Big Ben" class. Figure VI.3 (a) presents the output of the saliency detector, representing each salient point by a circle centered at the salient point location and with radius equal to the salient point scale. Note that while the regions of support of various salient points overlap with the target object, there are a number of false-positives. Figure VI.3 (b) presents all the salient points extracted from 40 training image after alignment by the displacements learned with EM. The points assigned to the true and false-positive classes are then presented in (c) and (d), respectively. The color coding reflects the posterior probability \(h_{ik}^+\) of assignment to the true-positive class (grayscale ranging from black for '0' to white for '1'). Finally, (e) and (f) present a visualization of the Gaussians, \(\Sigma^+\) and \(\Sigma^-\), that compose the configuration model. Note that the true-positive component has a shape that closely resembles that of the object of interest, while the false-positive component has broader support and, therefore, accounts for the salient points not associated with the target.

Figure VI.4 presents some examples of the classification of salient points into true and false-positives. Points depicted in white are true-positives (\(h_{ik}^+ > h_{ik}^-\)), while false-positives are shown in black.

It is worth noting that the model of (VI.11) could also account for scaling variations, by introducing a scaling factor \(g_k\) per image in (VI.11), i.e. multiplying
Figure VI.3  (a) Saliency detection result, (d) Init saliency points overlapped over 40 samples, (b) Overlapped points classified as ”object”, (c) Overlapped points classified as ”noise”, (e) Visualized variance, $\Sigma_1$, and (f) Visualized variance, $\Sigma_0$.

Figure VI.4  Saliency points classification results for some of training examples. White circle belongs more to ”object” class and black one belongs more to ”noise” class.
l_{ik} - t_k$ by $g_k$ and $s_{ik}$ by $g_k^2$. It would also be possible to assume that only the true-positive component of the configuration model is centered at the original, maintaining an arbitrary mean $\mu^-$ for the false-positive component in (VI.12). Our experience is that this makes the model too flexible. The false-positive salient points are usually clustered around the target (see e.g. Figure VI.3(a)) with some outliers points randomly located away from it. In the absence of the zero-mean constraint the false-positive component of the model is sometimes attracted by these outliers, and the false-positives that surround the target are incorrectly assigned to the true-positive component. In the remainder of this work we only consider the model of (VI.12).

### VI.D Generation of ROI mask

The configuration model of Figure VI.4 (e)-(f) can be seen as a template for the saliency configuration of the target object. Given an image on which saliency is to be determined, henceforth referred as the test image, a ROI mask is generated by finding the image location at which the saliency map best matches the template. In this section we describe a minimum probability of error rule to determine this location.

#### VI.D.1 Discriminant saliency

The salient point detection procedure of Section VI.B is first applied to the test image, and the saliency map approximated by a Gaussian mixture, as in (VI.10). To achieve invariance to the scale of the target, the test image is subject to a four-level Gaussian pyramid decomposition, and the procedure repeated at the four levels. The search for the best match to the saliency template is finally performed at all scales, and the scale with the best match is selected.
VI.D.2 Generative model

The generative model of Section VI.C.2 is extended to account for the matching operation. We introduce a variable $P$ which encodes the location of the template. Saliency is then defined locally, by restricting the saliency map of the test image to the region covered by the template, according to

$$P_{X|P}(x|p) = \sum_{i=0}^{1} P_{X|Y,P}(x|i, p) P_{Y|P}(i|p). \quad \text{(VI.19)}$$

The terms, $P_{Y|P}(i|p)$, model the prior dependence of saliency on image location. This denotes the probability salient regions being covered by the template. The terms $P_{X|Y,P}(x|i, p)$ model the restriction of the saliency to the template region. We adopt a soft weighting function which re-scales the mixture components of (VI.10) according to a Gaussian function that reflects the desired window structure. In particular,

$$P_{Y|P}(1|p) = \Pi_1 = \sum_{i=1}^{N} \pi_i \mathcal{G}(l_i, p, \Sigma_w) \quad \text{(VI.20)}$$

$$P_{Y|P}(0|p) = \Pi_0 = 1 - \Pi_1$$

$$P_{X|Y,P}(x|i, p) = \mathcal{G}(x - p, 0, \Sigma_i) \alpha_i$$

where $l_i$ and $s_i$ are the parameters of (VI.10),

$$\beta_i = \frac{\pi_i \mathcal{G}(l_i, p, \Sigma_w)}{\sum_j \pi_j \mathcal{G}(l_j, p, \Sigma_w)},$$

and $\Sigma_w = \frac{\Sigma^+ + \Sigma^-}{2}$.

VI.D.3 Optimal template location

To determine whether the localized saliency map of (VI.21) matches the configuration model of (VI.12), we rely on the Bayes decision rule. The best matching location, $p^*$, is the location which maximizes the posterior for the object class label given the location $p$ and saliency samples $x$. 
\[ p^* = \arg \max_p P_{Y|\mathbf{X},p}(1|\mathbf{x},p) \]

\[ = \arg \max_p \frac{P_{X|\mathbf{Y},P}(x|1,p)P_{Y|P}(1|p)}{\sum_{y=0}^1 P_{X|\mathbf{Y},P}(x|y,p)P_{Y|P}(y|p)} \]

\[ = \arg \max_p \log \frac{P_{X|\mathbf{Y},P}(x|1,p)P_{Y|P}(1|p)}{P_{X|\mathbf{Y},P}(x|0,p)P_{Y|P}(0|p)} \]

The quantity we need to find here is \( \log P_{X|Y,P}(\mathbf{x}|y,p) \). For one sample point \( \mathbf{x}_1 \),

\[ \log P_{X|Y,P}(\mathbf{x}|y,p) = \log G(\mathbf{x}_1 - \mathbf{p}, 0, \Sigma_y) \]

For \( K \) samples

\[ \log P_{X|Y,P}(\mathbf{x}|y,p) = \sum_{i=1}^K \log G(\mathbf{x}_i - \mathbf{p}, 0, \Sigma_y) \]

\( \mathbf{x} \) is the sample from the distribution of \( \sum_{i=1}^N \beta_i \int G(\mathbf{x}, l_i - \mathbf{p}, s_i l) \) therefore, for the infinite number of samples from this distribution, the log likelihood is considered as its expectation

\[
\begin{align*}
\log P_{X|Y,P}(\mathbf{x}|y,p) &= \sum_{i=1}^N \beta_i \int G(\mathbf{x}, l_i, s_i l) \log G(\mathbf{x} - \mathbf{p}, 0, \Sigma_y) d\mathbf{x} \\
&= \sum_{i=1}^N \beta_i \int G(\mathbf{x}, l_i - \mathbf{p}, s_i l) \log G(\mathbf{x}, 0, \Sigma_y) d\mathbf{x} \\
&= \sum_{i=1}^N \beta_i \left( - \log(2\pi) - \frac{1}{2} \log |\Sigma_y| \\
&\quad - \frac{1}{2} \text{trace}(\Sigma_y^{-1} s_i l) - (l_i - \mathbf{p})^T \Sigma_y^{-1} (l_i - \mathbf{p}) \right) \\
&= \sum_{i=1}^N \beta_i \left( - \log(2\pi) - \frac{1}{2} \log |\Sigma_y| - \frac{1}{2} \text{trace}(\Sigma_y^{-1}) \sum_{i=1}^N \beta_i s_i \\
&\quad - \sum_{i=1}^N \beta_i ((l_i - \mathbf{p})^T \Sigma_y^{-1} (l_i - \mathbf{p})) \right)
\end{align*}
\]
Therefore, equation (VI.21) becomes

\[
\begin{align*}
\mathbf{p}^* &= \arg\max_{\mathbf{p}} f(\mathbf{p}) \\
&= \arg\max_{\mathbf{p}} \{ \mathbb{E}_{\mathbf{x}|\mathbf{p}} [\log P_{\mathbf{x}|Y}(\mathbf{x}|1)|\mathbf{p} = \mathbf{p}] \} \\
&= \arg\max_{\mathbf{p}} \{ \mathbb{E}_{\mathbf{x}|\mathbf{p}} [\log P_{\mathbf{x}|Y}(\mathbf{x}|0)|\mathbf{p} = \mathbf{p}] \} \\
&+ \log P_Y(1|\mathbf{p}) - \log P_Y(0|\mathbf{p}) \} \\
&= \arg\max_{\mathbf{p}} \{ \sum_{i=1}^{N} \beta_i (s_i^T(\mathbf{l}_i - \mathbf{p})(\mathbf{l}_i - \mathbf{p})^T) \} \\
&+ \log \frac{\Pi_1}{\Pi_0} \\
&= \arg\max_{\mathbf{p}} \{ \sum_{i=1}^{N} \beta_i (s_i^T(\mathbf{l}_i - \mathbf{p})(\mathbf{l}_i - \mathbf{p})^T) \} \\
&+ \log \frac{\Pi_1}{\Pi_0} \\
\end{align*}
\]

In the case of clean training data which only has the perfectly aligned object, e.g. training examples given in UIUC car side database [4], the term for the background does not have any information i.e. $P_{\mathbf{x}|Y}(\mathbf{x}|0, \mathbf{p})$ has uniform distribution. Therefore, $f(\mathbf{p})$ becomes

\[
\begin{align*}
-\text{trace}\{\Sigma_i^{-1}\} \sum_{i=1}^{N} \beta_i (s_i^T(\mathbf{l}_i - \mathbf{p})(\mathbf{l}_i - \mathbf{p})^T) \} + \log \frac{\Pi_1}{\Pi_0}
\end{align*}
\]

VI.D.4 ROI mask

Given the optimal template location, the ROI mask is, once again, generated with recourse to decision-theoretic processing. The salient points are first assigned to the true and false-positives, through the computation of class-posterior probabilities. This is identical to the E-step of the hierarchical EM algorithm, and the posterior of class $t$ given the $i^{th}$ point is given by

\[
h_i^t = \frac{[\mathcal{G}(l_i, \mathbf{p}^*, \Sigma_j)e^{-\frac{1}{2}\text{trace}\{((\Sigma_j)^{-1}(s_i^T\mathbf{l})\})}]}{\sum_{t\in\{0,1\}}[\mathcal{G}(l_i, \mathbf{p}^*, \Sigma_i)e^{-\frac{1}{2}\text{trace}\{((\Sigma_i)^{-1}(s_i^T\mathbf{l})\})}]} \cdot (VI.24)
\]

The points are then weighed by their assignments to the true-positive class, and the Gaussian of maximum weighted likelihood computed as in the M-step of EM,

\[
\Sigma = \frac{\sum_{i=1}^{N} h_i^t \pi_i (l_i - \mathbf{p}^*)(l_i - \mathbf{p}^*)^T + s_i^2 \mathbf{l}}{\sum_{i=1}^{N} h_i^t \pi_i} \quad (VI.25)
\]
The posterior, point \( x \) being in class \( Y = 1 \), is proportional to the multiplication between observation of distribution \( x \) and prior distribution for the class \( Y = 1 \)

\[
P_{Y|X,P}(1|x,p^*) \propto P_{X|Y,P}(x|1,p^*)P_Y(p^*)
\]

\[
= \mathcal{G}(x,p^*,\Sigma_*)\mathcal{G}(x,p^*,\Sigma_1)
\]

\[
= \mathcal{G}(x,p^*,(\Sigma_*^{-1} + \Sigma_1^{-1})^{-1})
\]

\[
= \mathcal{G}(x,p^*,\Sigma_{**})
\]

where \( \Sigma_1 \) is the covariance for the class \( Y = 1 \) learned in the training stage.

Finally, the ROI mask is determined by thresholding this Gaussian.

\[
ROI_p = \{ x | g(x,p^*,\Sigma_{**}) > \alpha \}
\]

For the remaining \( f_1(p) \) after suppressing the area depicted as ROI for the first \( p_1^* \) the next maximum point can be found until

\[
\max_p f_n(p) < \beta f(p_1^*)
\]

Figure VI.5 shows this procedure with a specific example. (a) in Figure VI.5 is the original test image and (b) is the visualized saliency list for the object 'street sign'. Each point in the saliency list is represented by one Gaussian with mean of \( l_k \), variance of \( s_k I \), and peak amplitude of \( \pi_k \). Through this step the given test image is converted into saliency domain where the points matching the trained feature are highlighted. Then, the matchness with the configuration of saliency points are tested to find center of the object.

(c) in Figure VI.5 shows the probability map considering the configuration of saliency, \( f(p) \), and (d) is the ROI mask for the object of 'street sign'. The position with correct configuration is highlighted in saliency domain, by applying shape information for the object. The point with the maximum response and its neighboring positions, indicated by the scale of the object, constitute the final ROI mask.
Figure VI.5  (a) Original image, (b) Visualized saliency map, (c) Probability map, $P_l$, considering the configuration, (d) Final ROI mask for the object of 'street sign'.

Figure VI.6 shows more examples explaining the proposed algorithm for finding the ROI mask of the given object class. In this experiments the parameter $\alpha = 0.1$ and $\beta = 0.99$ were set as constant for all of the object classes.

VI.E  Preparing training set

Internet is a abundant source for examples to be used as training set for the detector but it also gives lots of garbages in terms of training examples because the current methods are based on text which is attached to each image. To find "right" images for the given object query, image filtering or sorting mechanism is needed. By comparing each image's saliency value which is supposed to be inside of object and is not normalized within images, each image has it's own score and can be sorted in terms of its relative saliency.

The overall flow is shown in Figure VI.8.

(1) Discriminant features are found which can distinguish the whole downloaded images vs. random image set. Here, the random image set is comprised with
Figure VI.6  Examples showing the process to find ROI mask, row 1 is for the object, “Big Ben”, row 2 for “Eiffel Tower”, row 3 for “Statue of Liberty”, and row 4 for “Tower of Pisa”. The first column shows the given test images, the second one shows visualised saliency lists, the third one clustered saliency points, the fourth one gray level of ROI mask, and the fifth column shows the binary ROI masks for given object.
Figure VI.7  First row is the result with UIUC car side database and the second row is the result with CalTech face database. (a,d) 1-precision vs Recall curve for UIUC database (b,e) ROC area curve (c,f) PSNR improvements by using ROI coding

images sampled in 200 image categories. 5 images are sampled in each category.

(2) After converting example images into feature response domain, the configuration parameters are learned explained in section VI.C.3

(3) Using the learned parameters, optimal template location, $p^*$, is found in each image. Refer section VI.D.3.

(4) Based on the center, $p^*$, and shape, $\Sigma_1$, information, score of each image is defined as $\sum_{i=1}^{N} S(l_i)(s_i^2)G(l_i, p^*, \Sigma_1)$, where $S(l_i)$ is the saliency amplitude for the location $l_i$ before within image normalization. So the score counts saliency amplitude located inside of object area.
Figure VI.8 Automatic image sorting process. Among downloaded images using google search engine, good example images are selected by sorting with saliency value.

VI.F Experimental results

VI.F.1 Scoring evaluation

To filter unwanted images among downloaded image set, each image is evaluated it's relevancy based on saliency inside object explained in section VI.E. In this section the performance of scoring is presented, which checks "right" image has high score.

Randomly selected 1,000 images from 200 image categories are used as negative examples and positive set are composed of two types; one is from the given object database so that we can assure that all of images are good examples and the other is from the web search so some part of those are good but the others are not relevant examples for the given query.

The given query in this experiment is "car" so we used UIUC car side test set database as the "for sure" one and downloaded 1,000 images from web with Flickr search engine under the query of "car". Image size was specified as "small" where longer edge is resized as 240 while the ratio is kept constant. The downloaded 1,000 images are labeled as 0, 1, or 2 manually, where 2 means the image is good example for learning "car", 1 means the image is not a good example though it has some part of cars, and 0 means the image is nothing related to "car". Among 1,000 images, 246 images are labeled as 2, 108 images are labeled as 1, and 646 images are labeled as 0. So basically 24.6% of images are useful for the training.
with downloaded images.

Figure VI.9 shows the score histogram distribution and ROC curve for three types of images. The first group is "for sure" database which is UIUC car side database so every image has at least one clear object. The second group is part of downloaded "car" images which is labeled as 2 and the third group is downloaded images labeled as 0 or 1. From Figure VI.9 (a) one can see that "for sure" images have higher score over web images and images labeled 2 have higher score over images labeled 0 or 1.

For the case of having threshold of 18 in Figure VI.9 (a), 168 images out of 170 total "for sure" set (98.82%) are classified as positive, 170 images out of 246 good examples among the downloaded images (69.11%) are classified as positive while 268 images out of 754 bad examples among the downloaded images (35.54%) are misclassified as positive.

Figure VI.9 (c) shows the true positive (TP) rate according to number of selected images. In the "unsorted image" case, images are selected from the one which comes up first after searching. As can be seen in this figure, the TP rate is higher if it includes images which have consistent object(s) (the sorted images case with "for sure and web images"). For the case of web images, sorted one keeps higher TP rate over unsorted case as number of images increase. For instance of 320 images, sorted case has 132 good examples and unsorted case has 70 good examples.

**VI.F.2 Object region detection**

To evaluate the proposed algorithm in a quantitative way, we applied the algorithm to the well known CalTech and UIUC car side databases where the groundtruths are given for "face" and "car side" respectively. The main task here is to localize the given object area.

Training was done with two different training sets; one is the training set given with the data and the other is the training set collected from web searching
under the search of "face" and "car". Comparing the results with other algorithms where the same databases were used, the performance and robustness in terms of training noise is shown.

1) Dedicated train set

In this case the training set is chosen from the given dataset.

The data set is randomly divided in half to produce the train and test sets. In the UIUC car side database, the original train set is used for training and the single scale original test set is used for testing.

2) Web train set

In this case training examples are collected from web search under the query of "face" and "car". The site used as the search engine is Flicker so as to limit the search to photos.

After downloading 1,000 images under one query, coarse salient feature and the coarse common configuration of saliency points are learned by using all of downloaded images as positive examples. With the learned coarse parameters, the detection algorithm is applied to the downloaded train examples and examples are sorted upon their $\max P f(p)$ in (VI.22). Among these, the top 100 examples were used as the training set. For the negative examples, 1,059 images collected from 200 different image categories are used. This universal negative set is applied as negative examples for all of the object categories shown in this paper.
Figure VI.10 Top row: clean training examples given with data set bottom row: selected training set from downloaded images through web search for the object "car".

Figure VI.10 shows training examples in two different sets for the object "car". Examples in the upper row are the taken from the data set which does not have any other object or background and the lower row shows the selected examples after downloading under the query of "car".

In terms of the metric that measures the performance, ROC area and PSNR curve corresponding to different Bit Per Pixel(BPP) are shown for each database. The True Positive(TP) and False Positive(FP) used in the ROC area curve is computed with the following equation

\[
TP = \frac{A_{det} \cap A_{truth}}{A_{truth}} \tag{VI.26}
\]

\[
FP = \frac{A_{det} \cap A_{c}^{truth}}{A_{c}^{truth}}
\]

where \(A_{det}\) is the detected object area and \(A_{truth}\) is the truth area for the object.

The ROC area curve is generated by varying the parameters \(\alpha\) and \(\beta\) in section VI.D and PSNR curve is obtained with one specific parameter pair which is set by cross validation.

Agarwal [4] also proposed a metric to classify points in the image. In this metric a point is classified as true if it falls within an ellipse centered at the groundtruth rectangle.

With precision recall curve, the exactness of matchness between local maximum and the actual center point of the object is measured and ROC area...
Figure VI.11  Example of typical multi object detection problem in a scene. Gaussian window can be overlapped between objects.

curve shows the coverage of detected area for the truth.

Figure VI.7 shows results for the UIUC car side database. (a) is the precision recall curve measured with the same method in [4], (b) is the ROC area curve, and (c) is the PSNR performance according to different bpp.

In this database there are two types of scenes; one is comprised of multiple cars and the other is comprised of only one car. Since we use a gaussian window when localising the search window to a specific position, if two objects are lie close together, the second local maximum tends to be tilted to the first maximum position unless the responses from the first object and those from the second objects are exactly same. Figure VI.11 shows one such example indicating this problem. In this example, the center of the second (right car) object is shifted to its left because of the first (left car) object fails to be classified as correct detection, i.e. it fails to be inside of the ellipse.

However, this is not a problem for finding "Region of Interest" areas. Thus the results are displayed exclusively for the case where is only a car in the image. An interesting point in our results is that the difference in performance for the two kinds of training set is not large considering the effort devoted to preparing these sets.

Also, as can be seen in (b) and (c), the difference is even smaller when the coverage is more concerned. In (c), "Uniform" is the case where thresholds,
Figure VI.12  First row shows images which are coded with normal jpeg and the bottom row shows the corresponding images coded with detected ROI mask. BPP is 0.2; 23.4Kbits for each example in “face” class and BPP is 0.6; 14.04Kbits for each example in “car side” class.

α and β, are set equally to all of test images and “Customized” is the ideal case where thresholds are customized for each test example. By using ROI coding, we can save 35.7% bits for the best to get the same quality of compressed images. The average image size in the database is 195 \times 120 and usually takes 20% of the images. (d), (e), and (f) in Figure VI.7 shows the result with CalTech face database. The size of used images is 298 \times 197 and usually 31.77% are taken by objects.

Note that the difference between dedicated training and web training is bigger in UIUC car data because the training data in UIUC is much cleaner than training set in Caltech face database which has clutter background. Even with web training and fixed threshold for car side one can still save 14.29% bits for the same quality. Also it is natural that ROI coding is more efficient with UIUC car data than Caltech face because the object is smaller comparing the whole image in “car” images.

Figure VI.12 shows the examples of the coded result for CalTech face/UIUC car side databases.
VI.G Possible applications

JPEG encoding with ROI information can unevenly distribute limited bandwidth in terms of what users want to see or keep more clearly, it can be used in most applications which suffers from low bit rate transmission or limited amount of memory spaces. In the experiments we consider the scenarios of transmitting through wireless channel which allows a limited bandwidth for each user or saving images within portable devices which has limited amount of memory. With ROI detection and ROI coding mechanism, bits needed for representing the image is totally controlled not only by simple image resolution or size but also by the contents. In this section we assumed possible scenarios and present ROI detection and JPEG lossy coding results corresponding to the detected ROI mask for the interesting object. The training set for each interesting object is collected from web and the size of each training set is 100 out of 1,000 original downloads. The top 100 examples are selected by sorting examples with respect to it’s maximum $f(p)$ in equation (VI.22).

VI.G.1 ‘street sign’ detection for OCR application

One of the possible applications can be found in OCR(Optical Character Recognition) for finding location from the street sign taken by cell phone. Sending related area information like nearby restaurants, bus stop, or famous places based on the picture of the place where users want to know is one of the possible application for cell phone service company, and street signs are the key to enable to place recognition. To recognize the characters in the sign, it should have quality high enough for recognition. Figure VI.13 shows coding examples with same Bit Per Pixel, and Table VI.1 shows examples having same quality for the interesting region but using different bits. The results shown in Table VI.1 demonstrate that if a user needs more space or bandwidth, he/she can reduce the size of the coded image to maximally one fourth of the original size without losing any quality for
Figure VI.13  Coding examples under the same BPP (a), (d), and (g) are the original images, (b), (e), and (h) are the normal JPEG case, and (c), (f), and (i) is ROI JPEG using the detected ROI mask.

the interesting region.

VI.G.2  Examples of famous monument detection

The other possible useful scenario for ROI coding can be for the travelers who tend to suffer from limited storage. Figure VI.14 shows coding examples for the ROI detection coding of famous monuments.
Figure VI.14  Coding examples under the same BPP. The first column is the images coded with normal JPEG, and the second column is the images of ROI JPEG using the detected ROI mask.
Table VI.1: Bits to achieve the same PSNR for two compression methods, ROI vs normal coding.

<table>
<thead>
<tr>
<th>Image Size</th>
<th>BPP (ROI)</th>
<th>BPP (Normal)</th>
<th>Total Bits (ROI)</th>
<th>Total Bits (Normal)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>338×270</td>
<td>0.298</td>
<td>0.529</td>
<td>27,195</td>
<td>48,277</td>
<td>0.563</td>
</tr>
<tr>
<td>353×265</td>
<td>0.294</td>
<td>0.593</td>
<td>27,502</td>
<td>55,472</td>
<td>0.496</td>
</tr>
<tr>
<td>448×336</td>
<td>0.21</td>
<td>0.533</td>
<td>31,611</td>
<td>80,231</td>
<td>0.394</td>
</tr>
<tr>
<td>329×466</td>
<td>0.291</td>
<td>0.503</td>
<td>44,614</td>
<td>77,117</td>
<td>0.579</td>
</tr>
<tr>
<td>756×567</td>
<td>0.14</td>
<td>0.307</td>
<td>60,011</td>
<td>131,596</td>
<td>0.456</td>
</tr>
<tr>
<td>1311×875</td>
<td>0.0525</td>
<td>0.209</td>
<td>60,224</td>
<td>239,749</td>
<td>0.251</td>
</tr>
<tr>
<td>1242×932</td>
<td>0.184</td>
<td>0.528</td>
<td>212,988</td>
<td>611,183</td>
<td>0.348</td>
</tr>
<tr>
<td>1366×932</td>
<td>0.0338</td>
<td>0.0961</td>
<td>43,031</td>
<td>122,346</td>
<td>0.352</td>
</tr>
<tr>
<td>1600×1195</td>
<td>0.03</td>
<td>0.071</td>
<td>57,360</td>
<td>135,752</td>
<td>0.423</td>
</tr>
<tr>
<td>1682×2243</td>
<td>0.0362</td>
<td>0.0577</td>
<td>136,573</td>
<td>217,686</td>
<td>0.627</td>
</tr>
</tbody>
</table>

VI.H Appendix

VI.H.1 Hierarchical mixture density estimation using EM iteration

The higher level mixture model, \( M_l = \{\alpha_j, \Sigma_j | j = 0, 1\} \), for saliency points is

\[
P(X) = \sum_{j=0}^{1} \alpha_j g(x, 0, \Sigma_j) \tag{VI.27}
\]

and the lower level mixture model, \( M_{l+1} = \{l_{ik}, \mu_k, s_{ik}, \pi_{ik} | i = 1, \ldots, N_k, k = 1, \ldots, K\} \) is

\[
P(X) = \sum_{k=1}^{K} P_k(X) \tag{VI.28}
\]

\[
P_k(X) = \sum_{i=1}^{N_k} \pi_{ik} g(x, l_{ik} - \mu_k, s_{ik}^2 I)
\]

Considering a virtual sample \( X = \{X_{ik} | i = 1, \ldots, N_k \text{ and } k = 1, \ldots, K\} \) with the size of \( M_{ik} = \pi_{ik} \times N \) from one of the components in the lower level model where \( N \) is the total number of virtual set, the likelihood for the virtual sample
under the model $M_l$ is

$$P(X_{ik}|z_{ik} = j, M_l) = \prod_{m=1}^{M_{ik}} g(x_{ikm}, l_{ik} - \mu_k, \Sigma_j) \quad (VI.29)$$

$$P_k(X_k|M_l) = \prod_{i=1}^{N_k} \prod_{j=0}^{1} \{\alpha_j P(X_{ik}|z_{ik} = j, M_l)\}^{\eta(z_{ik}=j)}$$

$$P(X|M_l) = \prod_{k=1}^{K} P_k(X_k|M_l)$$

where

$$\eta(z_{ik} = j) = \begin{cases} 1 & z_{ik} = j \\ 0 & otherwise \end{cases}$$

For the complete data set the log likelihood becomes

$$\log P(X|M_l) = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \sum_{j=0}^{1} \eta(z_{ik} = j) \log(\alpha_j P(X_{ik}|z_{ik} = j, M_l)) \quad (VI.30)$$

The key quantity, $P(X_{ik}|z_{ik} = j, M_l)$, is computed as presented in [162].

$$\log P(X_{ik}|z_{ik} = j, M_l) = M_{ik} \left[ \frac{1}{M_{ik}} \sum_{m=1}^{M_{ik}} \log P(x_{ikm}^m|z_{ik} = j, M_l) \right]$$

$$= M_{ik} E_{M_{i+1,ik}}[\log P(x|z_{ik} = j, M_l)]$$

$$= M_{ik} \left[ \log K_j - \frac{1}{2} E_{M_{i+1,ik}}[x^T \Sigma_j^{-1} x] \right]$$

$$= M_{ik} \left[ \log K_j - \frac{1}{2} (l_{ik} - \mu_k)^T \Sigma_j^{-1} (l_{ik} - \mu_k) \right.$$  

$$\left. - \frac{1}{2} \text{trace}(\Sigma_j^{-1} s^2_{ik} I) \right]$$

where $K_j = \frac{1}{\sqrt{(2\pi)^2 |\Sigma_j|}}$.

Therefore,

$$P(X_{ik}|z_{ik} = j, M_l) = \left[ g(l_{ik}, \mu_k, \Sigma_j) e^{\frac{1}{2} \text{trace}(\Sigma_j^{-1} s^2_{ik} I)} \right]^{M_{ik}}$$

$$\quad (VI.32)$$
The maximization of the m-step is carried out by maximizing the Lagrangian

\[
L = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \sum_{j=0}^{1} \{h_{ik}\}^j \times \log(\alpha_j P(X_{ik}|z_{ik} = j, M_l)) + \lambda(\sum_{j=0}^{1} \alpha_j - 1)
\]  

By setting the derivatives of \(L\) with respect to \(\alpha_j\), \(\lambda\), \(\mu_k\), and \(\Sigma_j^{-1}\), the result of m-step in equation (VI.14) is obtained.

\[\text{VI.I Acknowledgments}\]

The text of Chapter VI, in full, is based on the material as it appears in: Sunhyoung Han and Nuno Vasconcelos, “Object-based regions of interest for image compression”1, in *Data Compression Conference (DCC)*, 2008 and Sunhyoung Han and Nuno Vasconcelos, “Automatic detection of Object-based ROI (Region of Interest) for image compression”1, submitted to *IEEE Transactions on Image Processing*. The dissertation author was a primary researcher and an author of the cited material.
Chapter VII

Conclusions
In this thesis we have presented a novel hierarchical discriminant saliency network for object recognition and showed the generality of the proposed architecture by providing 1) biological plausibility to standard neurophysiological model of V1 2) flexibility to implement different saliency criterion including both bottom-up and top-down methods and 3) comparison to the structures of successful computer vision algorithms.

Under the assumptions of natural image statistics, the computation of discriminant saliency is completely consistent with the standard neural architecture in the primary visual cortex (V1). We have showed that meaningful statistical quantities can be computed through each and every biological hardware of lateral divisive operation, non-linearities, and pooling. It is then provided that a number of proposals for the measurement of visual saliency can be implemented by the proposed network. Among different saliency mechanisms, experimental results suggest the following points. 1) Subtle modifications to the saliency computations can lead to substantial changes of network behavior, 2) Top-down saliency has a significant positive impact on recognition but this impact is largest when saliency is discriminant (of a top-down nature). 3) Unsupervised learning of interest points does not perform as well, although it consistently achieves better performance than no saliency at all. 4) Max-based pooling does not appear to have an advantage over averaging indicating that selecting discriminant features is more important than locating them exactly. In fact, assigning semantics to all network components helps understand why network modifications that appear to be minor a-priori can have a dramatic impact in performance.

We also have shown that state-of-the-art recognition algorithms in the computer vision literature can actually be mapped to network form. This enables direct comparison by setting of comparable parameters the same. The results suggest that computer vision models achieve gains simply relying on extensions of computing units which is not necessarily related to model itself. With the comparable computing units discriminant saliency network architecture showed better
performance than state-of-the-art algorithms. The intrinsically different nature of discriminant saliency network is exploited in the amorphous object detection. The unique property of equating feature absence as a new feature makes the proposed network suitable for amorphous object detection where most of computer vision algorithms fail.
Bibliography


